Algorithmic Operations Research

Reducing the tongue–and–groove underdosage in MLC shape matrix decomposition

Thomas Kalinowski

Volume 3, Number 2, Fall 2008

URI: https://id.erudit.org/iderudit/aor3_2art05

See table of contents

Publisher(s)

Preeminent Academic Facets Inc.

ISSN

1718-3235 (digital)

Explore this journal

Cite this article

Kalinowski, T. (2008). Reducing the tongue–and–groove underdosage in MLC shape matrix decomposition. *Algorithmic Operations Research*, *3*(2), 165–174.

Article abstract

We present an algorithm for optimal step-and-shoot multileaf collimator field segmentation minimizing tongue-and-groove effects. Adapting the concepts of [7] we characterize the minimal decomposition time as the maximal weight of a path in a properly constructed weighted digraph. We also show that this decomposition time can be realized by a unidirectional plan, thus proving that the algorithm from [9] is monitor unit optimal in general and not only for unidirectional leaf movement. Our characterization of the minimal decomposition time has the advantage that it can be used to derive a heuristic for the reduction of the number of shape matrices following the ideas of [7].

Algorithmic Operations Research Vol.3 (2008) 165–174

Reducing the tongue–and–groove underdosage in MLC shape matrix decomposition

Thomas Kalinowski

Institut für Mathematik, Universität Rostock, 18051 Rostock, Germany

Abstract

We present an algorithm for optimal step-and-shoot multileaf collimator field segmentation minimizing tongue-and-groove effects. Adapting the concepts of [7] we characterize the minimal decomposition time as the maximal weight of a path in a properly constructed weighted digraph. We also show that this decomposition time can be realized by a unidirectional plan, thus proving that the algorithm from [9] is monitor unit optimal in general and not only for unidirectional leaf movement. Our characterization of the minimal decomposition time has the advantage that it can be used to derive a heuristic for the reduction of the number of shape matrices following the ideas of [7].

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

1. Introduction

An important method in cancer treatment is the use of high energetic radiation. In order to kill tumor cells the patient is exposed to radiation that is delivered by a linear accelerator whose beam head can be rotated about the treatment couch. Inevitably the healthy tissue surrounding the tumor is also exposed to some radiation. So the problem arises to arrange the treatment in a way such that the tumor receives a sufficiently high uniform dose while the damage to the normal tissue is as small as possible. The standard approach to this problem is as follows. First the patient body is discretized into so called *voxels*. The set of voxels is then partitioned into three sets: the clinical target volume, the critical structures and the remaining tissue. There are certain dose constraints for each of these parts. Basically the dose in the target volume has to be sufficient to kill the cancerous cells and the dose in the critical structures must not destroy the functionality of the corresponding organs. The determination of a combination of radiation fields is usually done by inverse methods based on certain physical models of how the radiation passes through a body. In the early 1990's the method of intensity modulated radiation therapy (IMRT) was developed in order to obtain additional flexibility. Using a multileaf collimator (MLC) it is possible to form homogeneous fields of different shapes. By superimposing some homogeneous fields an intensity modulated field is delivered. An MLC consists of two banks of metal leaves which block the radiation and can be shifted to form irregularly shaped beams (Fig. 1).

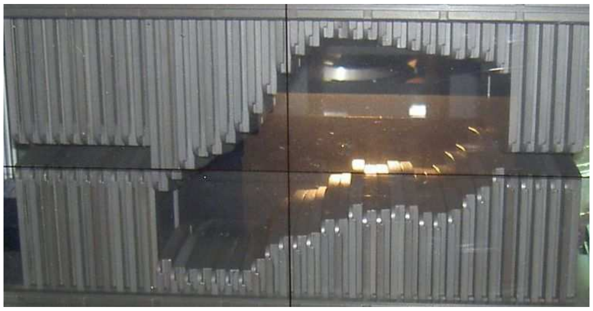

Fig. 1. The leaf pairs of a multileaf collimator (MLC)

The most common approach in treatment planning is to divide the optimization into two phases. At first, a set of beam angles and corresponding fluence matrices are determined. In a second step a sequence of leaf positions for the MLC for each of the angles is determined that yields the desired fluence distribution. Very recently there have been attempts to combine both steps into one optimization routine [5,12]

In this paper we concentrate on the second step, the shape matrix decomposition problem. Suppose we have fixed the beam angles from which the radiation is released, and for each of the beam angles we are given a fluence distribution that we want the patient to be exposed to. After discretizing the beam into *bixels* we can assume that the fluence distribution is given as a nonnegative integer $m \times n-$matrix A. Each row of the matrix corresponds to a pair of leaves of the MLC, and the entry a_{ij} represents the required fluence at bixel (i, j). When the MLC is used in the so called step–and–shoot mode the given fluence distribution is realized by superimposing a number of differently shaped homogeneous

Email: Thomas Kalinowski [thomas.kalinowski@uni-rostock.de].

fields coming from different combinations of the leaf positions. For example, Figure 2 shows a sequence of leaf positions for the matrix

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

where the shading indicates the region which is covered by the leaves.

The problem of realizing a given intensity matrix A leads to the problem of representing A as a positive integer combination of certain $(0,1)$–matrices, called shape matrices, which represent the possible leaf positions. So the realization in Fig. 2 corresponds to the decomposition in (1). In order to compare different decompositions of an intensity map we consider two quantities (where we adopt the terminology of [1]). For a decomposition $A = \sum_{t=1}^{k} u_t S^{(t)}$, the sum of the coefficients is proportional to the total irradiation time and is called *decomposition time*, $DT = \sum_{k=1}^{t} u_k$. The number t of used shape matrices, called *decomposition cardinality* (DC), influences the total treatment time due to the setup time between the delivery of different shapes. Our objectives in constructing a decomposition are to minimize both DT and DC. In this paper we consider two additional constraints that come from the technical restrictions in many of the available MLCs. The interleaf collision constraint (ICC) forbids the overlapping of opposite leaves in adjacent rows. Another restriction is due to the tongue–and–groove leaf arrangement of the MLCs (see Fig. 3).

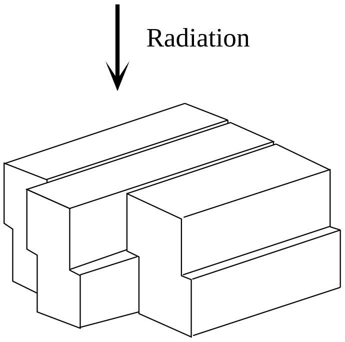

Fig. 3. The tongue–and–groove design of the leaves of an MLC.

There is a narrow strip in the border region between two adjacent rows that is covered by both leaves and this may lead to underdosage effects in these regions, as is illustrated in Figure 4 for the fluence matrix $A = \left(\begin{smallmatrix} 2 & 3 \\ 3 & 4 \end{smallmatrix}\right)$.

In order to minimize this effect we require that $a_{ij} \leq a_{i+1,j}$ implies that bixel $(i+1,j)$ is exposed whenever bixel (i,j) is exposed (similarly for $i-1$ instead of $i+1$). Thus we assure that the overlap region of two bixels always receives the smaller one of the relevant doses. We say that a shape matrix decomposition of A satisfies the tongue–and–groove constraint (TGC) if this condition holds for all used shape matrices. This intuitive concept of minimizing underdosage is made more precise in Lemma 1 below. Of course, when the total delivery time increases due to adding the TGC, the total leakage radiation through closed leaves also increases, so there might be a tradeoff between reduction of TG-underdosage and increasing leakage. But numerical experiments indicate that the increase of delivery time compared to the unconstrained case is rather small.

Starting with [3] and [6] several algorithms were proposed for the shape matrix decomposition problem [1,2,4,8,13,14]. Methods for eliminating the tongue–and–groove underdosage were presented in [9–11]. The algorithm from [9] is DT–optimal, as is shown for unidirectional plans in [9] and will be proved without restriction for the leaf movement direction in the present paper. Adapting the approach of [4], in [7] we characterized the minimal DT for the decomposition with ICC as the maximal weight of a path in a certain digraph. In this paper we further modify this approach such that the TGC is included. In addition, we present a greedy heuristic for the reduction of the number of shape matrices and present some numerical test results.

2. Mathematical formulation of the DT–decomposition problem with ICC and TGC

Throughout the rest of the paper, for a natural number n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$ and for natural numbers $m \leq n$, $[m,n]$ denotes the set $\{m, m+1, \ldots, n\}$. In this section we formulate the shape matrix decomposition problem and give a min–max–characterization of the optimal solution very similar to the one used in [7]. We start with a formal characterization of the shape matrices that are allowed in a decomposition of a given intensity matrix A.

Definition 1. Let A be an intensity matrix. An *A–shape matrix* is an $m \times n$-matrix $S = (s_{ij})$ with entries from $\{0,1\}$, such that there exist integers $l_i,\ r_i\ (i \in [m])$ with the following properties:

$$l_i < r_i \qquad (i \in [m]), \qquad (2)$$

$$s_{i,j} = \begin{cases} 1 & \text{if } l_i < j < r_i \\ 0 & \text{otherwise} \end{cases} \qquad (i \in [m],\ j \in [n]), \qquad (3)$$

$$(4)$$

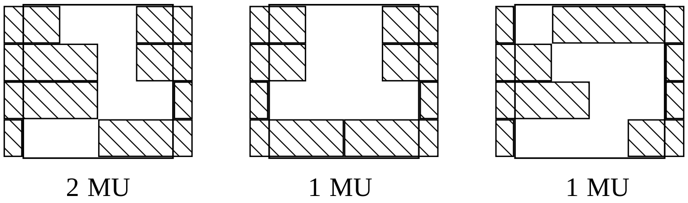

Fig. 2. A realization of the intensity matrix A using an MLC. The numbers below the leaf positions indicate the number of monitor units required.

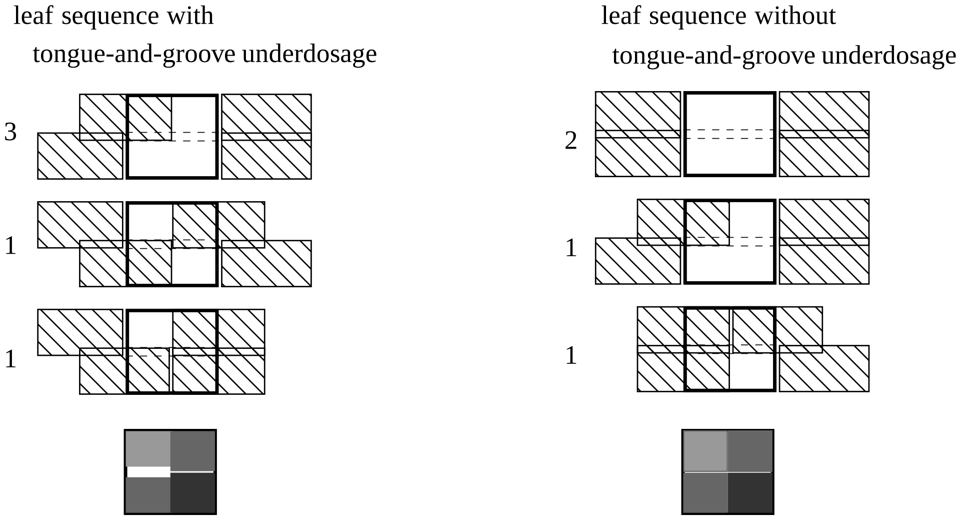

Fig. 4. Two different realizations of the same fluence matrix. The numbers next to the leaf positions indicate the irradiation times for the corresponding beams. In the left version the overlap between bixels $(1,1)$ and $(2,1)$ receives no radiation at all.

$$\text{ICC:}\quad l_i < r_{i+1},\ r_i > l_{i+1} \qquad (i \in [m-1]), \tag{5}$$

and we have

$$\text{TGC:}\begin{cases} a_{ij} \le a_{i+1,j} \wedge s_{ij} = 1 \Rightarrow \\ \qquad s_{i+1,j} = 1\ (i \in [m-1],\ j \in [n]), \\ a_{ij} \le a_{i-1,j} \wedge s_{ij} = 1 \Rightarrow \\ \qquad s_{i-1,j} = 1\ (i \in [2,m],\ j \in [n]). \end{cases} \tag{6}$$

A *shape matrix decomposition* of an intensity matrix A is a representation

$$A = \sum_{k=1}^{t} u_k S^{(k)} \tag{7}$$

with positive integers u_k and A–shape matrices $S^{(k)}$ ($k \in [t]$). The decomposition time (DT) of this decomposition is $\sum_{k=1}^{t} u_k$ and the shape matrix decomposition problem is to find, for given A, a shape matrix decomposition with minimal DT. We want to give a precise description of the sense in which condition (6) ensures that the TG-underdosage is minimized. For this purpose we define the tongue and groove error of a decomposition (7) at bixel (i,j) by

$$TG(i,j) = \min\{a_{ij}, a_{i+1,j}\} - \textstyle\sum_{k=1}^{t} u_k s_{ij}^{(k)} s_{i+1,j}^{(k)} \\ (i \in [m-1], j \in [n]).$$

The sum in the right hand side of this equation is the total fluence delivered to the overlap between rows i and $i+1$ in column j, because this overlap is open in the $k-$th shape if and only if $s_{ij}^{(k)} = s_{i+1,j}^{(k)} = 1$. This sum is at most $\min\{a_{ij}, a_{i+1,j}\}$:

$$a_{ij} = \sum_{k=1}^{t} u_k s_{ij}^{(k)} \ge \sum_{k=1}^{t} u_k s_{ij}^{(k)} s_{i+1,j}^{(k)},$$

and similarly for $a_{i+1,j}$. Thus $T(i,j) \ge 0$ and every positive value of $T(i,j)$ indicates an underdosage. The following lemma states that the underdosage is minimized for every (i,j) if all the shape matrices satisfy condition (6).

Lemma 1. *For a decomposition $A = \sum_{k=1}^{t} u_k S^{(k)}$, we have $TG(i,j) = 0$ for all $(i,j) \in [m-1] \times [n]$ if and only if every shape matrix $S^{(k)}$ satisfies (6).*

Proof. By symmetry, we may assume $a_{ij} \le a_{i+1,j}$. We obtain $T(i,j)=0$ if and only if

$$a_{ij}=\sum_{k=1}^{t}u_k s_{ij}^{(k)}=\sum_{k=1}^{t}u_k s_{ij}^{(k)}s_{i+1,j}^{(k)},$$

and this is the case if and only if $s_{i+1,j}^{(k)}=1$ whenever $s_{ij}^{(k)}=1$. □

In order to characterize the minimal DT we use a similar approach as in [7]. We construct a digraph $G=(V,E)$ as follows.

$$\begin{aligned}V&=\{0,1\}\cup([m]\times[0,n+1])\,,\\ E&=E_1\cup E_2\cup E_3\cup E_4\quad\text{where}\end{aligned}$$

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

Here 0 and 1 serve as starting and end point, respectively, and the vertices in $[m]\times[n]$ correspond to the entries of A. The two extra columns $[m]\times\{0\}$ and $[m]\times\{n+1\}$ have the purpose to simplify the notation: they assure that for every $(i,j)\in[m]\times[n]$ there are vertices $(i,j-1)$ and $(i,j+1)$. Without this, in several of the arguments below, it would be necessary to treat the first and the last column separately (then 0 and 1 would have to play the role of $(i,0)$ and $(i,n+1)$, respectively). To be able to treat the first and the n-th column exactly as the remaining columns, we also put $a_{i,0}=a_{i,n+1}=0$ ($i\in[m]$). We define the weight function $w:E\to\mathbb{Z}$:

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

Example 1. Figure 5 shows G corresponding to the matrix

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

The following theorem, which is proved in Sections 3. and 4., is the main result of this paper and the basis of the decomposition algorithm.

Theorem 1. *The minimal DT of a shape matrix decomposition of a nonnegative matrix A equals the maximal weight of a $(0,1)-$path in G.*

For convenience we denote this maximal weight by $c(A)$:

$$c(A)=\max\{w(P)\ :\ P\text{ is a }(0,1)-\text{path in }G\}.\quad(8)$$

Observe that the results from [4] and [7] can be seen as characterizations of the minimal DT in terms of maximal path weights for different variants of the problem corresponding to manfacturer specific restrictions.

- MLC without restriction of leaf movement: use the graph G without the vertical arcs.
- MLC with interleaf collision but without tongue and groove: use the same graph G, but with modified weights for the vertical arcs.

So the only case that cannot be treated in this framework is an MLC with tongue and groove and without interleaf collision.

3. The lower bound

In this section we show that the maximal weight of a $(0,1)-$path in G is a lower bound for the DT of a decomposition of A, thus proving the first half of Theorem 1. The basic idea of the proof is a combination of the arguments in [1] and [9], the main difference to [9] being that we do not require the leaf sequence to be unidirectional. For our argument below we need an exact description of how the numbers

$$\alpha(i,j):=\max\{w(P)\ :\ P\text{ is a }(0,(i,j))-\text{path in }G\}$$

can be computed. This description is given in Algorithm 1.

The underlying principle can be described as follows. We proceed columnwise. Assuming we have already determined the values in column $j-1$ we initialize column j with $\alpha(i,j):=\alpha(i,j-1)+w((i,j-1),(i,j))$. After that we modify these values in order to satisfy the conditions

$$\begin{aligned}\alpha(i,j)&\geq\alpha(i-1,j)+w((i-1,j),(i,j))\\ &\qquad\text{for }i\in[2,m],\\ \alpha(i,j)&\geq\alpha(i+1,j)+w((i+1,j),(i,j))\\ &\qquad\text{for }i\in[m-1].\end{aligned}$$

Now the statement of the following lemma is obvious.

Lemma 2. *Algorithm 1 computes the numbers $\alpha(i,j)$ in time $O(m^2n)$.*

Suppose $A=\sum_{k=1}^{t}S^{(k)}$ is a shape matrix decomposition of A. We characterize the shape matrix $S^{(k)}$ by its left and right leaf positions $l_i^{(k)}$ and $r_i^{(k)}$ ($i\in[m]$). For $(i,j)\in[m]\times[n+1]$, let L_{ij} denote the set of indices k with $l_i^{(k)}<j$, and similarly, let R_{ij} denote the

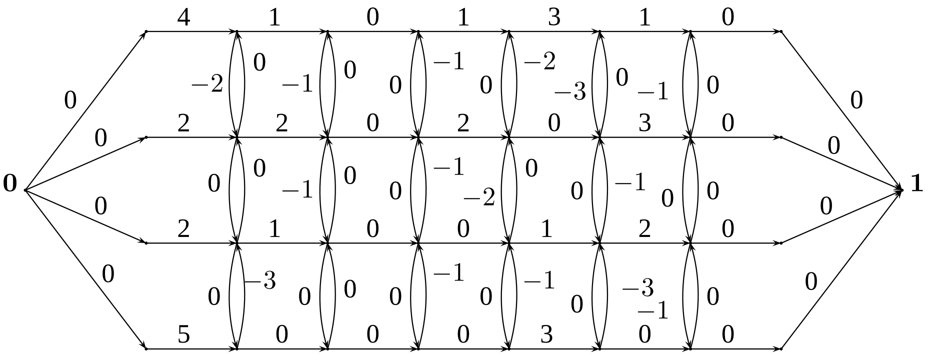

Fig. 5. The digraph G corresponding to matrix A.

Algorithm 1 (Computation of the numbers $\alpha(i,j)$**).**

```
for $i = 1,\dots,m$ do $\alpha(i,1) := a_{i1}$
for $j = 2,\dots,n+1$ do
  for $i = 1,\dots,m$ do $\alpha(i,j) := \alpha(i,j-1) + w((i,j-1),(i,j))$
  for $i = 2,\dots,m$ do
    if $\alpha(i,j) < \alpha(i-1,j) + w((i-1,j),(i,j))$ then
      $\alpha(i,j) := \alpha(i-1,j) + w((i-1,j),(i,j))$
    if $\alpha(i-1,j) < \alpha(ij) + w((i+1,j),(i,j))$ then Update$(i-1)$
```

```
Function Update$(k)$
$\alpha(k,j) := \alpha(k+1,j) + w((k+1,j),(i,j))$
if $k \geq 2$ and $\alpha(k-1,j) < \alpha(kj) + w((k,j),(k-1,j))$ then Update$(k-1)$
```

set of indices k with $r_i^{(k)} \leq j$. More formally,

$$L_{ij} = \{k \in [t] \;:\; l_i^{(k)} < j\},$$

$$R_{ij} = \{k \in [t] \;:\; r_i^{(k)} \leq j\}.$$

Then $|L_{in}|$ is the number of shape matrices which contribute to row i, and $\max_{i\in[m]} |L_{in}|$ is a lower bound for the DT. In the next lemma we collect some simple observations about the sets L_{ij} and R_{ij}.

Lemma 3. *(1) For $(i,j) \in [m]\times[n]$, $R_{ij} \subseteq L_{ij}$ and $|L_{ij} \setminus R_{ij}| = a_{ij}$.*

(2) For $(i,j) \in [m]\times[n]$, $|L_{ij}| \geq |L_{i,j-1}| + \max\{0, a_{ij} - a_{i,j-1}\}$.

(3) For $(i,j) \in [2,m]\times[n]$, $R_{i-1,j} \subseteq L_{ij}$ and $R_{ij} \subseteq L_{i-1,j}$.

(4) For $(i,j) \in [2,m]\times[n]$,

$$a_{i-1,j} \leq a_{ij} \Longrightarrow L_{i-1,j} \setminus R_{i-1,j} \subseteq L_{ij} \setminus R_{ij}$$
$$a_{i-1,j} \geq a_{ij} \Longrightarrow L_{i-1,j} \setminus R_{i-1,j} \supseteq L_{ij} \setminus R_{ij}$$

Proof. The first statement is a simple consequence of the facts that $r_i^{(k)} \leq j$ implies $l_i^{(k)} < j$ and that $s_{ij}^{(k)} = 1$ if and only if $k \in L_{ij} \setminus R_{ij}$. The second statement is clear if $a_{ij} \leq a_{i,j-1}$, since $L_{i,j-1} \subseteq L_{ij}$. If $a_{ij} > a_{i,j-1}$, there must be at least $a_{ij} - a_{i,j-1}$ shape matrices $S^{(k)}$ with $s_{ij}^{(k)} = 1$ and $s_{i,j-1}^{(k)} = 0$. For these shape matrices we have $l_i^{(k)} = j-1$, so $k \in L_{ij} \setminus L_{i,j-1}$ and this proves the second claim. Using the ICC we obtain the first inclusion in the third statement:

$$k \in R_{i-1,j} \implies r_{i-1}^{(k)} \leq j \implies l_i^{(k)} < j \implies k \in L_{ij},$$

and similarly the second one. For the fourth statement, assume $a_{i-1,j} \leq a_{ij}$. Using the TGC we obtain

$$k \in L_{i-1,j} \setminus R_{i-1,j} \implies s_{i-1,j}^{(k)} = 1 \implies s_{ij}^{(k)} = 1$$
$$\implies k \in L_{ij} \setminus R_{ij}.$$

This gives the first implication, and the second one is proved similarly. □

Next, we show that the numbers $\alpha_1(i,j)$ bound the cardinalities $|L_{ij}|$ from below.

Lemma 4. *For $(i,j) \in [m]\times[n]$, we have $\alpha(i,j) \leq |L_{ij}|$.*

Proof. We proceed by induction. For $j = 1$, $\alpha(i,1) = a_{i1}$ and the claim is obvious, since we need at least a_{i1} shape matrices with $l_i^{(k)} = 0$. Suppose the statement of the lemma is false, and let j be the index of the first column where, for some row i, we have $\alpha(i,j) > |L_{ij}|$. From Lemma 3 we get

$$|L_{ij}| \geq |L_{i,j-1}| + \max\{0, a_{ij} - a_{i,j-1}\}$$
$$\geq \alpha(i,j-1) + w((i,j-1),(i,j)).$$

Hence after the initialization of column j in Algorithm 1 (line 3), we still have $\alpha(i,j) \leq |L_{ij}|$ for all $i \in [m]$.

Now let i be the index of the row where the claim of the lemma is violated for the first time when the algorithm is running. Consider this first violation and assume it occurs in line 6 of Algorithm 1. The case that it occurs in the function `Update`(k) is treated analogously.

Case 1. $a_{i-1,j} \leq a_{ij}$. In this case $w((i-1,j),(i,j)) = 0$, hence the updating step of the algorithm is $\alpha(i,j) := \alpha(i-1,j)$. By ***(iii)*** and ***(iv)*** in Lemma 3 we have

$$R_{i-1,j} \subseteq L_{ij} \text{ and } L_{i-1,j} \setminus R_{i-1,j} \subseteq L_{ij}.$$

Hence $L_{i-1,j} \subseteq L_{ij}$, and consequently $\alpha(i,j) = \alpha(i-1,j) \leq |L_{ij}|$, contradicting the assumption that the step leads to a violation of the claim.

Case 2. $a_{i-1,j} > a_{ij}$. Now the considered step is $\alpha(ij) := \alpha(i-1,j) - (a_{i-1,j} - a_{ij})$. Again by ***(iii)*** and ***(iv)*** from Lemma 3,

$$R_{i-1,j} \subseteq L_{ij} \text{ and } L_{ij} \setminus R_{ij} \subseteq L_{i-1,j} \setminus R_{i-1,j}.$$

This implies (using ***(i)*** from Lemma 3)

$$\begin{aligned}|L_{ij}| &\geq |R_{i-1,j}| + |L_{ij} \setminus R_{ij}|\\ &= (|L_{i-1,j}| - a_{i-1,j}) + a_{ij}\\ &\geq \alpha(i-1,j) - a_{i-1,j} + a_{ij} = \alpha(i,j),\end{aligned}$$

contradicting the assumption. □

Lemma 4 shows that the numbers $\alpha(i,n)$ ($i \in [m]$) are lower bounds for the DT. We state this conclusion as a lemma.

Lemma 5. *For any shape matrix decomposition of an intensity matrix A, we have*

$$DT \geq \max_{i\in[m]} \alpha(i,n) = c(A).$$

4. The algorithm

We compute a shape matrix decomposition of A according to Algorithm 2. This is essentially a reformulation of the algorithm of Kamath et al. [9], but we need it in this form in order to show that our characterization of the minimal DT in Theorem 1 is correct.

Algorithm 2 (DT-optimal shape matrix decomposition)**.**

```
for $k = 1, \ldots, c(A)$ do
    for $i = 1, \ldots, m$ do
        $l_i^{(k)} := \max\{j \in [0,n] \ : \ \alpha(i,j) < k$ or $j = n\}$
        $r_i^{(k)} := \min\{j \in [n+1] \ : \ \alpha(i,j) \geq k + a_{ij}$ or
                        $j = n+1\}$
    for $(i,j) \in [m] \times [n]$ do
        $s_{ij}^{(k)} := \begin{cases} 1 & \text{if } l_i^{(k)} < j < r_i^{(k)} \\ 0 & \text{otherwise} \end{cases}$
```

Lemma 6. *From Algorithm 2 we obtain a shape matrix decomposition of A with $DT = c(A)$.*

Proof. Clearly, the DT of the sum of shape matrices returned by the algorithm is $c(A)$. We divide the proof of the theorem into three parts.

Claim 1. The matrices $S^{(k)}$ form indeed a decomposition of A, that means $A = \sum_{k=1}^{c(A)} S^{(k)}$.
Fix some $(i,j) \in [m] \times [n]$. We have

$$\left(l_i^{(k)} < j \iff \alpha(i,j) \geq k\right) \text{ and}$$

$$\left(r_i^{(k)} > j \iff \alpha(i,j) < k + a_{ij}\right).$$

Together we obtain $s_{ij}^{(k)} = 1 \iff \alpha(i,j) - a_{ij} < k \leq \alpha(i,j)$, hence $\sum_{k=1}^{c(A)} s_{ij}^{(k)} = a_{ij}$, and this proves the claim.

Claim 2. The matrices $S^{(k)}$ satisfy the ICC.
Assume the claim is false. That means, for some $k \in [c(A)]$ and $i \in [m-1]$, $l_i^{(k)} \geq r_{i+1}^{(k)}$ or $r_i^{(k)} \leq l_{i+1}^{(k)}$. We consider only the first case, since the second one can be treated similarly. We put $j = r_{i+1}^{(k)}$. By construction and our assumption, we have

$$\alpha(i,j) < k \qquad \text{and} \quad \alpha(i+1,j) \geq k + a_{i+1,j}.$$

But on the other hand,

$$\begin{aligned}\alpha(i,j) &\geq \alpha(i+1,j) + w((i+1,j),(i,j))\\ &= \alpha(i+1,j) + \min\{0, a_{ij} - a_{i+1,j}\},\end{aligned}$$

thus $\alpha(i,j) \geq k$, and this contradiction proves the claim.

Claim 3. The matrices $S^{(k)}$ satisfy the TGC.
Suppose $a_{ij} \leq a_{i+1,j}$ and $s_{ij}^{(k)} = 1$, or equivalently $l_i^{(k)} < j < r_i^{(k)}$. By construction, this implies

$$k \leq \alpha(i,j) < k + a_{ij}. \tag{9}$$

Observe, that

$$w((i,j),(i+1,j)) = 0 \text{ and}$$

$$w((i+1,j),(i,j)) = a_{ij} - a_{i+1,j},$$

since $a_{ij} \leq a_{i+1,j}$. Using (9), we obtain the bounds

$$\begin{aligned}\alpha(i+1,j) &\geq \alpha(i,j) + w((i,j),(i+1,j))\\ &= \alpha(i,j) \geq k \text{ and}\end{aligned}$$

$$\begin{aligned}&k + a_{ij} > \alpha(i,j) \geq \alpha(i+1,j) + w((i+1,j),(i,j))\\ &k + a_{ij} > \alpha(i+1,j) + (a_{ij} - a_{i+1,j}).\end{aligned}$$

Hence $k \leq \alpha(i+1,j) < k + a_{i+1,j}$, and according to Algorithm 2, $s_{i+1,j}^{(k)} = 1$. Thus the first TGC is satisfied, and the second one is proved similarly. □

Together, Lemmas 5 and 6 prove Theorem 1.

5. Minimizing the number of shape matrices

The problem of minimizing the number of shape matrices is NP–hard even for a single row intensity matrix [1]. So it is natural to look for a heuristic approach that yields decompositions with a small number of shape matrices within a reasonable time even if optimality is not always reached. In [7] we used a greedy strategy in order to find a decomposition with minimal DT and a small number of shape matrices for MLCs with ICC but neglecting the TGC. This method can be modified to respect the TGC. In order to characterize the maximal coefficient u for which there is an A-shape matrix S, such that uS can be a term in a DT-optimal decomposition of A, we need the following lemma.

Lemma 7. *Let $A = \sum_{k=1}^{t} u_k S^{(k)}$ be a decomposition of A (i.e. the $S^{(k)}$ are A–shape matrices), and put $A^{(0)} = A$ and $A^{(k)} = A - \sum_{k'=1}^{k} u_{k'} S^{(k')}$ for $k \in [t]$. Then, for every $k \in [t]$ we have*

- $s_{ij}^{(k)} = 1$ *and* $s_{i+1,j}^{(k)} = 0 \;\Rightarrow\; a_{ij}^{(k-1)} \geq a_{i+1,j}^{(k-1)} + u \quad (i \in [m-1],\ j \in [n])$,
- $s_{ij}^{(k)} = 1$ *and* $s_{i-1,j}^{(k)} = 0 \;\Rightarrow\; a_{ij}^{(k-1)} \geq a_{i-1,j}^{(k-1)} + u \quad (i \in [2,m],\ j \in [n])$.

Informally speaking, if we consider the sequence of matrices starting with A and subtracting one by one the $S^{(k)}$ taking $S^{(k)}$ exactly u_k times, the lemma claims that in each step we subtract an A'–shape matrix, where A' is the resulting matrix after the previous step.

Proof. Assume the contrary and let k be the first index where one of the two claims fails to be true. By symmetry, we assume

$$s_{ij}^{(k)} = 1, \quad s_{i+1,j}^{(k)} = 0, \quad a_{ij}^{(k-1)} < a_{i+1,j}^{(k-1)} + u.$$

Since $S^{(k)}$ is an A–shape matrix, the TGC implies $a_{ij} > a_{i+1,j}$. From our assumption we obtain $a_{ij}^{(k)} < a_{i+1,j}^{(k)}$, hence

$$s_{ij}^{(k')} = 0 \quad \text{and} \quad s_{i+1,j}^{(k')} = 1$$

for some $k' > k$, contradicting the assumption that $S^{(k')}$ is an A–segment. □

We call a pair (u, S) of a positive integer u and an A–shape matrix S an *admissible segmentation pair*, if

- $A - uS$ is nonnegative,
- $s_{ij} = 1$ and $s_{i+1,j} = 0 \;\Rightarrow\; a_{ij} \geq a_{i+1,j} + u \quad (i \in [m-1],\ j \in [n])$,
- $s_{ij} = 1$ and $s_{i-1,j} = 0 \;\Rightarrow\; a_{ij} \geq a_{i-1,j} + u \quad (i \in [2,m],\ j \in [n])$,
- $c(A - uS) = c(A) - u$.

Now we proceed exactly as in [7]: we find an admissible segmentation pair (u, S) with maximal u and continue with $A - uS$ until we reach the zero matrix. In order to derive an upper bound for the coefficient u in an admissible segmentation pair (u, S), we use an idea from [2] and identify the set of segments with the set of paths from D to D' in the layered digraph $\Gamma = (W, F)$, constructed as follows. The vertices in the i−th layer correspond to the possible leaf positions in row i $(1 \leq i \leq m)$ and two additional vertices D and D' are added:

$$W = \{(i,l,r) : i \in [m],\ l \in [0,n],\ r \in [l+1, \ldots, n+1]\} \cup \{D, D'\}.$$

Between two vertices (i,l,r) and $(i+1,l',r')$ there is an arc if the corresponding leaf positions are consistent with the ICC, i.e. if $l' < r$ and $r' > l$. In addition, the arc set F contains all arcs from D to the first layer and from the last layer m to D', so

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

$$F_+(D) = \{(D,(1,l,r)) \;:\; (1,l,r) \in W\},$$
$$F_-(D) = \{((m,l,r),D') \;:\; (m,l,r) \in W\},$$
$$F_+(i) = \{((i,l,r),(i+1,l',r')) \;:\; l' < r,\ r' > l\}.$$

There is a bijection between the possible leaf positions and the paths from D to D' in Γ. This is illustrated in Fig. 6 which shows the paths in Γ for $m = 4$, $n = 2$, corresponding to the shape matrices

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \text{ (straight lines) and } \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ (dashed lines).}$$

For each vertex (i,l,r) let $u_0(i,l,r)$ denote an upper bound for the coefficient in an admissible segmentation pair (u, S) where S is a shape matrix with $l_i = l$ and $r_i = r$. Then any admissible segmentation pair (u, S) corresponds to a path

$$D, (1,l_1,r_1), (2,l_2,r_2), \ldots, (m,l_m,r_m), D'$$

with the following properties.

- For $i \in [m]$, $u_0(i,l_i,r_i) \geq u$.
- For $i \in [m-1]$ and $j \in [n]$,

$$l_i < j \leq l_{i+1} \text{ or } r_{i+1} \leq j < r_i \Longrightarrow a_{ij} \geq a_{i+1,j} + u,$$
$$l_{i+1} < j \leq l_i \text{ or } r_i \leq j < r_{i+1} \Longrightarrow a_{i+1,j} \geq a_{ij} + u.$$

If we have good upper bounds $u_0(i,l,r)$, this yields a considerable reduction of the set of shape matrices that have to be considered in the search for an admissible segmentation pair. In our implementation we used the bound from the following lemma.

Lemma 8. *For $i \in [m]$, let $g_i = c(A) - \sum_{j=1}^{n} \max\{0, a_{ij} - a_{i,j-1}\}$, and suppose (u, S) is an admissible segmentation pair with parameters l_i, r_i $(i \in [m])$. Then*

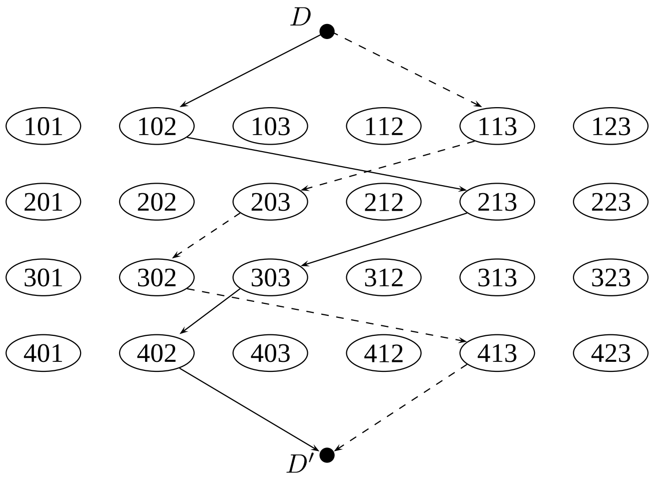

Fig. 6. The vertices of Γ for $m = 4$, $n = 2$ and two (D, D')–paths.

for $i \in [m]$,

$$u \le g_i \textit{ if } r_i = l_i + 1 \qquad (10)$$

$$u \le \min\{g_i + \max\{0, a_{i,r-1} - a_{ir}\}, g_i + \max\{0, a_{i,l+1} - a_{il}\}, \tfrac{1}{2}(g_i + \max\{0, a_{i,l+1} - a_{il}\} + \max\{0, a_{i,r-1} - a_{ir}\})\} \textit{ if } r_i > l_i + 1. \qquad (11)$$

Proof. For brevity of notation, let $d_{ij} = \max\{0, a_{ij} - a_{i,j-1}\}$ for $(i,j) \in [m] \times [n]$. Observe that $\sum_{j=1}^{n} d_{ij}$ is just the weight of the path

$$0, (i, 0), (i, 1), \ldots, (i, n), (i, n+1), 1$$

in G. The fact that (u, S) is an admissible segmentation pair implies,

$$\sum_{j=1}^{n} d'_{ij} \le c(A) - u, \qquad (12)$$

where $A' = (a'_{ij}) = A - uS$ and $d'_{ij} = \max\{0, a'_{ij} - a'_{i,j-1}\}$. If $r_i = l_i + 1$, $a'_{ij} = a_{ij}$ for all j and this implies (10). For (11), observe that

$$d'_{i,l_i+1} = d_{i,l_i+1} - \min\{u, d_{i,l_i+1}\},$$
$$d'_{i,r_i} = d_{i,r_i} + \max\{0, u - \max\{0, a_{i,r_i-1} - a_{ir_i}\}\},$$
$$d'_{ij} = d_{ij} \text{ for } j \notin \{l_{i+1}, r_i\}.$$

With (12) we obtain

$$\sum_{j=1}^{n} d_{ij} - \min\{u, d_{i,l_i+1}\}\} + \max\{0, u - \max\{0, a_{i,r_i-1} - a_{ir_i}\}\}\} \le c(A) - u,$$

hence

$$u - \min\{u, d_{i,l_i+1}\} + \max\{0, u - \max\{0, a_{i,r_i-1} - a_{ir_i}\}\} \le g_i$$

and this implies (11). □

Algorithm 3 summarizes our greedy approach for the construction of a DT-optimal shape matrix decomposition with a small DC.

6. Test results

We implemented Algorithm 3 in C++ and computed decompositions for $15 \times 15-$matrices, where the entries are chosen uniformly and independently from $\{0, \ldots, L\}$. Table 2 shows the results for different values of L, where for each row of the table we averaged over 1000 sample matrices. In the second column we have the average DT, which is the same as for the algorithm of Kamath et al. [9]. The third column shows the DC of a decomposition according to Algorithm 2 (or equivalently the algorithm of Kamath et al.). Clearly, this algorithm just aims at minimizing the DT without taking the DC into account, hence the DC almost equals the DT. In the fourth column we have the DC of the decompositions according to Algorithm 3, and we see that this approach yields considerable savings in terms of the number of used shape matrices. The CPU times (on a 2GHz workstation with 2GB RAM) in the third columns show that the algorithm is practicable for intensity matrices of the considered size (note that the times are for the decomposition of 1000 matrices, so the average time for a single matrix is still below

Table 1

L	DT	DC	DC	CPU time
		(plain)	(reduced)	(sec)
4	21.2	21.0	18.0	93
7	34.9	34.2	24.1	276
10	48.2	46.3	28.1	399
13	61.7	57.9	31.2	556
16	74.8	68.2	33.5	647

Test results for random 15×15-matrices with entries from $\{0, \ldots, L\}$.

Algorithm 3 (DT-optimal shape matrix decomposition with reduced DC)**.**

```
while $A \neq 0$ do
  Determine the complexity $c(A)$ and the numbers $u_0(i,l,r)$ for
  $i \in [m]$, $l \in [0,n]$, $r \in [n+1]$ according to Lemma 8
  $u := \max\{k$ : There is a path $P$ from $D$ to $D'$ in $\Gamma$
                      with $u_0(i,l,r) \geq k$ for all $(i,l,r) \in P\}$
  complete:=false;
  while (not complete) do
    for the paths $P$ in $\Gamma$ with $u_0(i,l,r) \geq k$ for all $(i,l,r) \in P$ do
      if (not complete) then
        Let $S$ be the shape matrix corresponding to $P$
        if $(u,S)$ is an admissible segmentation pair then
          complete:=true
    if (not complete) $u := u - 1$
  $A := A - S$
```

a second). But of course the backtracking for determining the maximal value of u becomes very slow for larger matrices, and more efficient methods are needed for matrix dimensions of practical relevance.

In order to evaluate the influence of the TGC, in Table 3 we compare results for different types of constraints.

Table 2

L	unconstrained		only ICC		ICC and TGC	
	DT	DC	DT	DC	DT	DC
4	17.9	10.9	19.5	14.5	21.2	18.0
7	29.5	13.1	31.7	18.2	34.9	24.1
10	40.9	14.7	43.8	20.7	48.2	28.1
13	52.4	15.8	55.7	22.5	61.7	31.2
16	63.8	16.8	67.7	24.0	74.8	33.5

Test results for random 15×15*-matrices with entries from* $\{0,\dots,L\}$ *for different types of constraints.*

Finally, we also tested our algorithm with 13 clinical matrices, each with 10 fluence levels. The results are shown in Table 4. The computation times for these matrices were negligible (less than a second).

Clearly, the addition of the TGC causes an increase in the DT and in the DC. Further investigations are necessary in order to evaluate the potential tradeoff between DT (and corresponding leakage) and tongue and groove underdosage.

Table 3

		unconstrained		with ICC		with ICC and TGC	
no.	size	DT	DC	DT	DC	DT	DC
1	10×11	16	8	16	8	17	11
2	10×9	16	7	16	8	19	13
3	9×9	20	8	20	10	20	12
4	9×9	19	8	19	11	21	15
5	10×8	15	7	18	9	19	11
6	9×9	17	9	17	9	19	11
7	10×8	18	7	18	10	21	12
8	14×12	22	9	22	10	25	14
9	14×10	26	10	30	15	34	19
10	14×10	22	9	23	13	28	15
11	15×10	22	10	22	11	25	16
12	15×11	23	10	23	12	23	16
13	14×10	23	9	24	11	27	17

Test results for clinical matrices.

7. Conclusion

We have presented an algorithm for MLC shape matrix decomposition taking into account the interleaf collision constraint and eliminating tongue-and-groove underdosage effects. We proved that our algorithm is optimal with respect to the total number of monitor units, thus completing the argument of [9] where the optimality was proved only for unidirectional schedules. In addition, we derived a heuristic approach to the reduction of the number of shape matrices. Two open questions arise immediately and are the subject of ongoing research. 1. Is there a nice characterization for the minimal decomposition time if we have no interleaf constraint but still want to eliminate tongue-and-groove underdosage? 2. What about a computationally more efficient heuristic for the decomposition cardinality?

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] 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] K. Engel. A new algorithm for optimal multileaf

collimator field segmentation. *Discrete Appl. Math.*, 152(1-3):35–51, 2005.

[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] 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.

[9] 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.

[10] S. Luan, C. Wang, D.Z. Chen, X.S. Hu, S. A. Naqvi, X. Wu, and C.X. Yu. An improved MLC segmentation algorithm and software for step-and-shoot IMRT delivery without tongue-and-groove error. *Med. Phys.*, 33(5):1199–1212, 2006.

[11] W. Que, J. Kung, and J. Dai. ‘Tongue–and–groove’ effect in intensity modulated radiotherapy with static multileaf collimator fields. *Phys. Med. Biol.*, 49:399–405, 2004.

[12] 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.

[13] 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.

[14] 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 7 March 2007; revised 16 June 2007; accepted 17 August 2007