Algorithmic Operations Research Vol.5 (2010) 119–132

Binary matrix decompositions without tongue-and-groove underdosage for radiation therapy planning ⋆

Antje Kiesel [a]

[a]Institute for Mathematics, University of Rostock, 18051 Rostock, Germany.

Céline Engelbeen [b]

[b]Department of Mathematics, Université Libre de Bruxelles CP 216, Boulevard du Triomphe, 1050 Brussels, Belgium.

Abstract

In the present paper we consider a particular case of the segmentation problem *arising in the elaboration of radiation therapy plans. This problem consists in decomposing an integer matrix A into a nonnegative integer linear combination of some particular binary matrices called* segments *which represent fields that are deliverable with a* multileaf collimator. *For the radiation therapy context, it is desirable to find a decomposition that minimizes the* beam-on time, *that is the sum of the coefficients of the decomposition. Here we investigate a variant of this minimization problem with an additional constraint on the segments, called the* tongue-and-groove constraint. *Although this minimization problem under the condition that the used segments have to respect the* tongue-and-groove *constraint has already been studied, the complexity of it is still unknown. Here we prove that in the particular case where A is a binary matrix this problem is polynomially solvable. We provide a polynomial procedure that finds such a decomposition with minimal beam-on time. Furthermore, we show that the beam-on time of an optimal decomposition (but not the segmentation itself) can be found by determining the chromatic number of a related perfect graph.*

Key words: intensity modulated radiation therapy (IMRT), consecutive ones property, tongue-and-groove constraint.

1. Introduction

Radiation therapy is one of the most prescribed treatment methods to cure cancer tumors. Its aim is to destroy the cells of the tumor by exposing it to radiation. But because we cannot isolate the tumor from the rest of the body, we have to take care of the healthy tissues and organs close to the tumor (called the *organs at risk*). Nowadays, most radiation therapy centers use a linear accelerator to send the radiation from different directions, which allows to deliver a higher dosage in the tumor than in the organs at risk. Moreover, a multileaf collimator (MLC, see Figure 1) is commonly used to cover a certain part of the radiation beam. This device consists of several pairs of metallic leaves which can block the radiation.

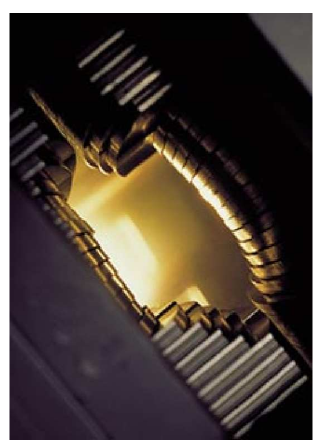


Fig. 1. The multileaf collimator (MLC).

⋆ This research is supported by an "Actions de Recherche Concertées" (ARC) project of the "Communauté française de Belgique". Céline Engelbeen is a research fellow of the "Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture" (FRIA) and Antje Kiesel is a research fellow of the German National Academic Foundation.

Email: Antje Kiesel [antje.kiesel@uni-rostock.de], Céline Engelbeen [cengelbe@ulb.ac.be].

The elaboration of a radiation therapy plan is commonly done in three steps.

(1) Different radiation angles are fixed, in such a way that the tumor is in the radiation epicenter and the organs at risk are protected as much as possible [9].
(2) For each direction, an intensity function is determined. This function is encoded as a nonnegative integer matrix A of size $m \times n$ where each entry represents an elementary part of the radiation beam (called *bixel*). The value of each entry corresponds to the intensity of radiation we want to send through that bixel.
(3) Each intensity matrix is segmented since the linear accelerator can only send a uniform radiation. This *segmentation step* consists in finding a sequence of positions of the MLC leaves.

In this paper we focus on the third step. Therefore our work is to find a set of positions of the leaves of the MLC in such a way that the dosage corresponding to the intensity matrix is finally delivered. We restrict our work to the case where the MLC is used in the so called *step-and-shoot* mode, which means that the leaves are never moving while the patient is irradiated.

Mathematically, the segmentation step amounts to decomposing matrix A into a nonnegative integer linear combination of some binary matrices whose shape can be reproduced by the leaves of the MLC (these binary matrices are called *segments*, see Figure 2).

For each row of these segments, the MLC has a left and a right leaf. The radiation is not capable of passing through bixels that are covered by a leaf, and so only passes through bixels located between the two leaves. This is why a segment is a binary matrix which has to satisfy the *consecutive ones property*, which means that the ones have to be grouped in a single block, in each row.

Throughout the paper, $[k]$ denotes the set $\{1, 2, \ldots, k\}$ for an integer k, and $[\ell, r]$ denotes the set $\{\ell, \ell+1, \ldots, r\}$ for integers ℓ and r with $\ell \leqslant r$. We also allow $\ell = r+1$ where $[\ell, r] = \emptyset$. Thus, an $m \times n$ matrix $S = (s_{ij})$ is a segment iff there are integral intervals $[\ell_i, r_i]$ for $i \in [m]$ such that

$$s_{ij} = \begin{cases} 1 & \text{if } j \in [\ell_i, r_i], \\ 0 & \text{otherwise.} \end{cases}$$

Hence, let p be the total number of segments and let $\mathcal{S} := \{S_1, \ldots, S_p\}$ be the whole set of segments of size $m \times n$. We look for nonnegative integers $u_1, \ldots, u_p$ such that

$$A = \sum_{t=1}^{p} u_t S_t.$$

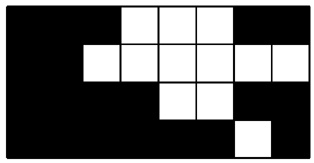
In the literature two different objective functions are considered (see [7] for a survey).

The *beam-on time problem* (BOT) consists in finding a decomposition of the matrix A which minimizes the sum of the coefficients $\sum_{t=1}^{p} u_t$. This corresponds to the goal, that the irradiation time for a patient should be minimized in order to decrease some undesirable effects like radiation leakage (a very small portion of the radiation is transmitted through the leaves). This problem is known to be polynomial and efficient methods for solving it have been proposed by several authors [1,2,8,13,14,16,22,23]. In some of the given references, also constrained versions of the beam-on time problem are considered. In clinical applications a lot of constraints may arise that reduce the number of deliverable segments. For some technical or dosimetric reasons we might look for decompositions where only a subset of all segments is allowed. Those subsets might be explicitely given [11] or defined by constraints like the interleaf collision constraint (also called interleaf motion constraint or interdigitation constraint, [2,3,13,14]), the interleaf distance constraint [10], the tongue-and-groove constraint [4,15,16,17,18,21], the minimum field size constraint [19] or the minimum separation constraint [11,16].

The *cardinality problem* (CP) consists in finding a decomposition of the given matrix A which minimizes the number of used segments, i.e. the cardinality of the support of all strictly positive coordinates of the vector $\mathbf{u}$, in order to decrease the duration of the radiation therapy session. This problem is known to be NP-hard [5], even if the matrix has only one row [2] or one column [6].

In the present paper we focus on finding a decomposition whose segments satisfy the *tongue-and-groove* constraint. The leaves are designed in such a way that the radiation cannot pass between two adjacent leaves. Therefore, there is a small overlap between adjacent leaves as illustrated in Figure 3.

To avoid under- and overdosage caused by this tongue-and-groove design, it is desirable to simultaneously irradiate the bixels of the same column as much as possible. Mathematically, this corresponds to the following two constraints:



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

Fig. 2. The leaves of the MLC determine a binary matrix called a segment

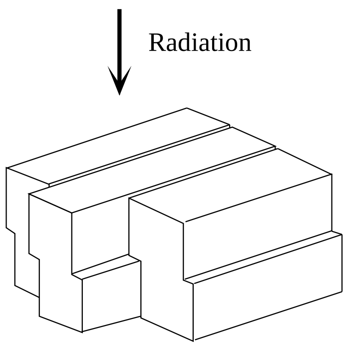


Fig. 3. The special design of the leaves of the MLC which leads to the tongue-and-groove constraint.

$$a_{ij} \leqslant a_{i-1,j} \text{ and } s_{ij} = 1 \Longrightarrow s_{i-1,j} = 1$$
$$a_{ij} \geqslant a_{i-1,j} \text{ and } s_{i-1,j} = 1 \Longrightarrow s_{ij} = 1.$$

This means that bixel $(i-1, j)$ and bixel (i, j) have to be irradiated exactly $\min\{a_{i-1,j}, a_{ij}\}$ times simultaneously. We call a segment satisfying the tongue-and-groove constraint a *TG-segment*. A decomposition of A into TG-segments is called *TG-decomposition*.

For the segmentation without tongue-and-groove underdosage, Kamath, Sahni, Palta, Ranka and Li [17,18] presented an algorithm which is beam-on time optimal in the special case of unidirectional leaf movements, i.e. when the leaves only move from left to right when their positions are changed from one segment to the next. But this does not lead to the optimal beam-on-time for the general case, as one can find decompositions with lower beam-on time where the leaves have to move both to the left and to the right during the treatment. Furthermore, we know polynomial algorithms for the case where both the tongue-and-groove and the interleaf collision constraint are taken into account [14]. In this case we again have an optimal unidirectional schedule, which is much easier to compute. The general problem can be formulated as network flow problem with side constraints using similar ideas as in [3] and thus solved by solving an integer linear problem. But the complexity of the general problem is still unknown. Luan, Wang, Chen, Hu, Naqvi, Wu and Yu have provided an approach which gives a decomposition of A which minimizes the number of used segments and the tongue-and-groove error [20].

In this paper, we restrict ourselves to decompositions of binary intensity matrices A. Obviously, we have $u_t = 1$ for all TG-segments S_t arising in the segmentation. Thus, minimizing the beam-on time is equivalent to minimizing the cardinality in our case. First, we prove that the problem under the tongue-and-groove constraint can be solved in polynomial time and provide an $O(m^2n^2)$ time algorithm to find such a TG-decomposition. Then we show that finding the optimal beam-on time of a TG-decomposition (but not the decomposition itself) can be done by relating the problem to a coloring problem in a perfect graph, which gives an alternative proof for the polynomiality of the problem.

2. TG-decompositions of binary matrices

For simplicity of notation, we add a 0-th and an $(n+1)$-th column to A and put $a_{i0} = a_{i,n+1} = 0$ for all $i \in [m]$. Similarly, we add a 0-th and an $(m+1)$-th row to A and put $a_{0j} = a_{m+1,j} = 0$ for all $j \in [0, n+1]$.

We use the well-known results for unconstrained decompositions of integer matrices. From [8] we know that the minimal beam-on time of a decomposition of the i-th row of A is

$$c_i(A) := \sum_{j=1}^{n} \max\{0, a_{ij} - a_{i-1,j}\} \tag{1}$$

which is in the binary case equal to the number of blocks of ones in row i. The minimal beam-on time of a decomposition of the whole matrix A is

$$c(A) := \max_{i \in [m]} c_i(A). \tag{2}$$

In the unconstrained case, the single row decompositions can be combined arbitrarily to form segments.

This is no longer the case if the tongue-and-groove constraint is considered. Obviously, if the input matrix is binary the tongue-and-groove-constraint reduces to the fact that consecutive ones in a column have to be irradiated simultaneously.

Definition 1.[Box] For $i_1, i_2 \in [m], i_1 \leq i_2$ and $j \in [n]$, the set of bixels $B = \{(i,j) \mid i_1 \leq i \leq i_2\}$ is called a *box*, if $a_{ij} = 1$ for all $(i,j) \in B$ and $a_{i_1-1,j} = a_{i_2+1,j} = 0$. If $a_{ij} = 1$ for $i \in [m]$, $j \in [n]$, we denote the unique box containing (i,j) by B_{ij}. If $a_{ij} = 0$, we define $B_{ij} = \emptyset$. Let the set of all boxes be $\mathcal{B}$.

In the unconstrained case, a decomposition of a binary matrix corresponds to a partition of the set of ones such that each subset forms a segment. Including the tongue-and-groove constraint, a decomposition is a partition of the set of boxes such that each subset has the consecutive ones property. Figure 4 shows an example for a decomposition of the set of boxes into MLC-segments.

We say that two boxes $B = [i_1, i_2] \times \{j\}$ and $B' = [i'_1, i'_2] \times \{j+1\}$ are *neighboring* if $[i_1, i_2] \cap [i'_1, i'_2] \neq \emptyset$. In such a case, the two boxes form a connected region of ones. Sometimes we have to separate these two boxes in order to satisfy the consecutive ones property of connected regions of ones. This is why we now introduce a splitting procedure on the set of boxes. For this we need some notation and use a geometrical point of view. We define the split $s_{B,B'}$ as the vertical line where the two boxes overlap, i.e.

$$s_{B,B'} := ([i_1, i_2] \cap [i'_1, i'_2]) \times \{j\}.$$

If we insert a split $s_{B,B'}$ between the two boxes B and B', B and B' do not form a connected region of ones anymore and we are not allowed to put both of them into the same segment. Here we say that the split is in position j because we split between column j and $j+1$. With each set of splits $\mathcal{SP}$, we associate a graph that models the connectedness of the ones in the matrix with respect to the given splits. Let $G = (V, E)$ be defined as follows:

$$\begin{aligned} V &= \{(i,j) \mid a_{ij} = 1\} \\ E &= \{\{(i,j),(i+1,j)\} \mid (i,j),(i+1,j) \in V\} \\ &\quad \cup \{\{(i,j),(i,j+1)\} \mid (i,j),(i,j+1) \in V, \\ &\quad \nexists\, s \in \mathcal{SP} : (i,j) \in s\} \end{aligned}$$

We call a subset of boxes $\mathcal{B}' \subseteq \mathcal{B}$ connected with respect to the split set $\mathcal{SP}$ if the subgraph induced by $\bigcup_{B \in \mathcal{B}'} B$ is connected. For each box B, its connected region is the connected component in the graph that contains B.

Definition 2.[Boxes of row i and number of splits] Let $i \in [m]$ be fixed. The boxes of row i are the elements of the set $\{B_{ij} \mid j \in [n]\}$. The *number of splits* of row i is defined as the minimal number of splits between neighboring boxes of row i that are necessary to make all the connected subsets of $\bigcup_{j \in [n]} B_{ij}$ satisfy the consecutive ones property. This number is denoted by $s_i(A)$.

So, the aim is to insert a split when a connected region of ones does not satisfy the consecutive ones property. Obviously, a connected subset of boxes from row i does not satisfy the consecutive ones property if and only if it contains a subset of the form $\{B_{i,j}, B_{i,j+1}, \ldots, B_{i,j'}\}$ with $j' > j+1$ and

$$\begin{aligned} &a_{i',j} = 1,\ a_{i',\ell} = 0 \text{ for all } \ell \in [j+1, j'-1] \\ &a_{i',j'} = 1,\ a_{k,\ell} = 1 \text{ for all } \ell \in [j, j'],\ k \in [i'+1, i] \end{aligned}$$

for some $i' < i$ or

$$\begin{aligned} &a_{k,\ell} = 1 \text{ for all } \ell \in [j, j'],\ k \in [i, i'-1] \\ &a_{i',j} = 1,\ \ a_{i',\ell} = 0 \text{ for all } \ell \in [j+1, j'-1] \\ &a_{i',j'} = 1 \end{aligned}$$

for some $i' > i$. We call the set $\{B_{i,j}, \ldots, B_{i,j'}\}$ an *i-cup* in the first case and an *i-cap* in the second case, as the zeros can be crossed below or above via other rows of ones. The situation is illustrated in Figure 5. If we talk about i-caps or i-cups, we call them i-obstacles. The comprised zero entries $a_{i',\ell}$ for $j+1 \leq \ell \leq j'-1$ are called *critical zeros*, as they destroy the consecutive ones property of the corresponding boxes and imply the necessity of a split.

For each row $i \in [m]$ and each i-obstacle $\{B_{ij}, \ldots, B_{ij'}\}$, we get an integral interval of possible split positions $[j, j'-1]$. At least one of these splits has to be chosen in order to destroy the i-obstacle and make the connected boxes satisfy the consecutive ones property. Let therefore $K^i_1 = [k_1, k'_1 - 1]^i, K^i_2 = [k_2, k'_2 - 1]^i, \ldots, K^i_{v_i} = [k_{v_i}, k'_{v_i} - 1]^i$ be the integral split intervals for all i-cups (ordered from left to right) and analogously let $L^i_1 = [\ell_1, \ell'_1 - 1]^i, L^i_2 = [\ell_2, \ell'_2 - 1]^i, \ldots, L^i_{w_i} = [\ell_{w_i}, \ell'_{w_i} - 1]^i$ be the integral split intervals for all i-caps (ordered from left to right). Here, v_i is the number of i-cups and w_i is the number of i-caps. Obviously, the K^i_j are pairwise disjoint and the L^i_j are pairwise disjoint for fixed i. Thus, for all

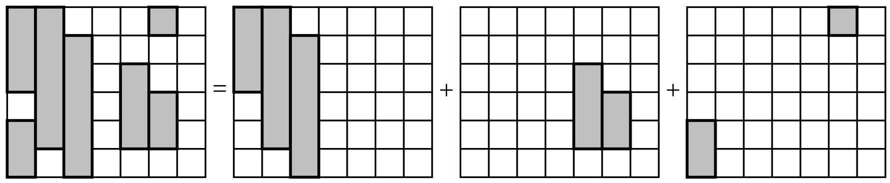


Fig. 4. Decomposition of the boxes into TG-segments.

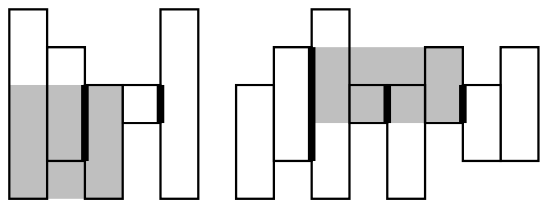


Fig. 5. Number of splits in row $i \in [m]$. The grey areas indicate an i-cap (left) and an i-cup (right). The thick lines indicate the splits that destroy the i-obstacles. Sometimes, one split can destroy two i-obstacles (one i-cap and one i-cup) as indicated by the second split on the right.

possible split positions $j \in [n-1]$, j can be contained in at most two of the intervals from above. As $s_i(A)$ is the minimal number of splits needed to destroy all i-obstacles, the computation of $s_i(A)$ amounts to finding a subset $M \subseteq [n-1]$ such that:

$$M \cap K_j^i \neq \emptyset \text{ for all } j \in [v_i]$$
$$M \cap L_j^i \neq \emptyset \text{ for all } j \in [w_i]$$
$$|M| \to \min$$

The optimal value of the objective function is $s_i(A)$.

This problem aims at partitioning the vertices of the corresponding interval graph into a minimum number of cliques (for more details see e.g. [12]). It can easily be solved by taking all the intervals of a row from the left to the right, and insert a split in the last possible position, that is the last position for which otherwise there would be an unsplit interval. Figure 6 gives a possible optimal solution for the problem, where the arrows indicate the splits given by this procedure.

Definition 3.[Tongue-and-groove complexity] We define the *TG row complexity* of row i by

$$c_i^{TG}(A) = c_i(A) + s_i(A).$$

The *TG complexity* of the intensity matrix A is defined by

$$c^{TG}(A) = \max_{i \in [m]} c_i^{TG}(A).$$

Our aim is to show that $c^{TG}(A)$ is the minimal beam-on time of a segmentation of the binary matrix A into TG-segments. For this, we need some more notation and lemmas. Obviously, $c_i^{TG}(A)$ is the minimal number of TG-segments we need to decompose the boxes of row i and $c^{TG}(A)$ is a lower bound for the minimal beam-on time.

Let us now assume that we have given A together with a set of splits $\mathcal{SP}$. If there exists some $s \in \mathcal{SP}$ with $(i,j) \in s$, then we do not allow to put the bixel (i,j) and $(i,j+1)$ into the same segment. We now generalize the definition of $c_i^{TG}(A, \mathcal{SP})$ and define it as the minimum number of segments that are necessary to decompose the set of boxes of row i with respect to the split set $\mathcal{SP}$. Obviously, if $\mathcal{SP} = \emptyset$, this corresponds to our previous definition of $c_i^{TG}(A)$. The i-obstacles and the corresponding split intervals for all $i \in [m]$ are also defined with respect to $\mathcal{SP}$, i.e. including splits reduces the number of i-obstacles.

If we insert a split between neighboring boxes B and B' and the split affects row i, there are two cases:

- The split increases the TG row complexity of row i (it can increase by at most one unit).
- The split does not increase the TG row complexity of row i.

If a split increases the TG row complexity of any row i, we call the split i-*infeasible*. Otherwise, the split is called i-*feasible*. A split is *feasible*, if it is i-feasible for all $i \in [m]$. For example, for the matrix

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

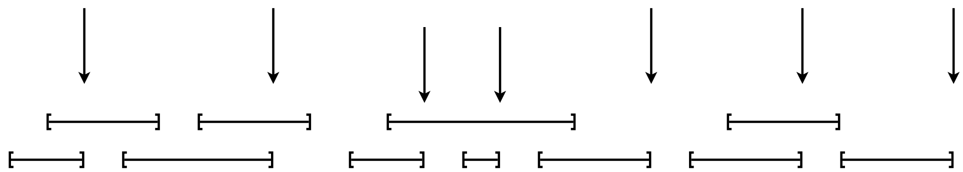


Fig. 6. Possible splits.

we have

$c_1^{TG}(A,\emptyset) = 4 + 0 = 4$
$c_2^{TG}(A,\emptyset) = 2 + 2 = 4$
$c_3^{TG}(A,\emptyset) = 3 + 0 = 3.$

Indeed, $s_1(A) = s_3(A) = 0$ since there are neither 1- or 3-cups nor 1- or 3-caps and $s_2(A)$ is equal to 2 since we need at least a split to destroy the split interval $[1, 2]$ and another one to destroy the split intervals $[6, 7]$ and $[7, 8]$. Notice that the split $s_{B_{2,2},B_{2,3}}$ is infeasible since it is 3-infeasible. Indeed, if we insert this split the number of blocks of ones in row 3 would be equal to 4 and hence $c_3^{TG}(A, \mathcal{SP})$ would increase. The split $s_{B_{2,1},B_{2,2}}$ is feasible. Similarly, the split $s_{B_{2,6},B_{2,7}}$ is 2-infeasible as only the 2-cup is destroyed while the remaining 2-cap requires a further split. The split $s_{B_{2,7},B_{2,8}}$ destroys both i-obstacles, does not increase the TG row complexity of row 2 and thus is 2-feasible (and also feasible).

The next lemma is easy to verify as it follows directly from the definition of the i-caps and i-cups.

Lemma 1. *Let row $k \in [m]$ have a k-cap (respectively k-cup) with split interval $[j, j'-1]$ and critical zeros in row $i > k$ (respectively $i < k$). Then all rows ℓ with $k \leq \ell < i$ (respectively $i < \ell \leq k$) also have the ℓ-cap (respectively ℓ-cup) with split interval $[j, j'-1]$.*

The next lemma follows from the previous one.

Lemma 2. *a) Let $i < i'$ such that there is an i-cap and an i'-cap with split interval $[j, j'-1]$ and the same critical zeros. Then every i'-cup with split interval $[\ell, \ell'-1]$ such that $[j, j'-1] \cap [\ell, \ell'-1] \neq \emptyset$ is also an i-cup.*

b) Let $i' < i$ such that there is an i-cup and an i'-cup with split interval $[j, j'-1]$ and the same critical zeros. Then every i'-cap with split interval $[\ell, \ell'-1]$ such that $[j, j'-1] \cap [\ell, \ell'-1] \neq \emptyset$ is also an i-cap.

Proof. We only prove a), as b) then follows by symmetry. For a), let $\{B_{ij}, \ldots, B_{ij'}\} = \{B_{i'j}, \ldots, B_{i'j'}\}$ be the i-cap and the i'-cap as in the following example:

$$\begin{array}{cc} & \begin{array}{ccccc} j & & & & j' \end{array} \\ \begin{array}{c} \\ i \\ \\ i' \\ \\ \end{array} & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Let $\{B_{i'\ell}, \ldots, B_{i'\ell'}\}$ be the i'-cup with $\ell < j'$ and $\ell' > j$. Thus, its critical zeros are in a row $k < i$ and by Lemma 1 $\{B_{i\ell}, \ldots, B_{i\ell'}\}$ is also an i-cup. ☐

Before we can prove the next lemma, we need some more notation. To clarify the notions, we introduce here the notations in terms of the split intervals as well as in terms of interval graphs. So, let the split intervals of some row $i \in [m]$ be ordered such that for consecutive intervals $I = [i_1, i_2]$ and $J = [j_1, j_2]$ $i_1 \leq j_1$ and if $i_1 = j_1$ then $i_2 \geq j_2$ holds. This means , if two intervals start at the same position, the longer one comes first with respect to this order. We associate with these intervals the interval graph $G_i(V, E)$. The set of vertices V includes a vertex I for each split interval of the row i and two vertices I and J are connected if the two corresponding split intervals have a non-empty intersection. Let us notice that $G_i(V, E)$ is a forest.

A set of split intervals $(I_1, \ldots, I_k)$ forms a *sequence* of row i if $I = I_1 \cup \cdots \cup I_k$ is a connected interval. The corresponding vertices $I_1, \ldots, I_k$ form a set of connected vertices of G_i. Such a set is a *component* if it cannot be extended, which means that $I_1, \ldots, I_k$ form a connected component of G_i. Finally, a *trunk* is a sequence $(I_1, \ldots, I_{\tilde{k}})$ with $\tilde{k} \leq k$ that has the property that there is no interval in the component $(I_1, \ldots, I_k)$ that is contained in any of the intervals $I_1, \ldots, I_{\tilde{k}}$. Note that a trunk consists of a set of split intervals corresponding to alternating i-caps and i-cups. The definitions are illustrated in Figure 7 and 8.

Obviously, for split intervals I and J, if $I \subseteq J$ then every split in I automatically also splits J. Thus, for

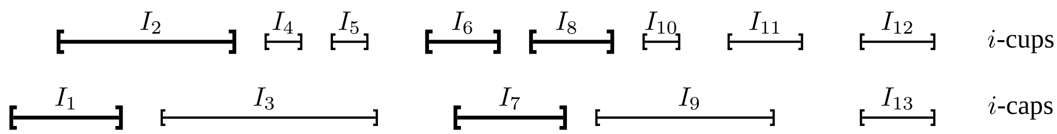


Fig. 7. Components and trunks. The intervals $I_1, \ldots, I_{13}$ are the split intervals of some row i. They decompose into three components of split intervals $(I_1, \ldots, I_5), (I_6, \ldots, I_{11})$ and (I_{12}, I_{13}). The trunks (I_1, I_2) and (I_6, I_7, I_8) are highlighted with bold lines. The trunk of the last component is empty.

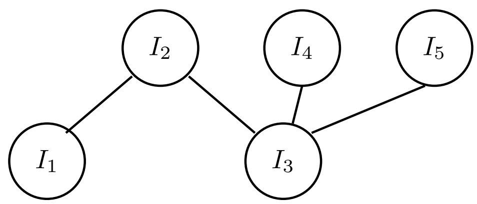


Fig. 8. The first component of the interval graph corresponding to the intervals $I_1, \ldots, I_5$ from Figure 7.

a component $(I_1, \ldots, I_k)$ in row i, the decision if a split in $I_1 \setminus I_2$ is i-feasible only depends on the trunk. Hence, for the first component in Figure 7, we see that each split of I_4 and I_5 will also split I_3. Therefore, because I_3 will automatically be split by the split we will have to insert in I_4, we do not have to care of that interval and the decision about the feasibility of a split in $I_1 \setminus I_2$ only depends on the trunk (I_1, I_2). As the number of intervals in this trunk is even, a split in $I_1 \setminus I_2$ is infeasible. The next lemma is obvious using the interval graphs.

Lemma 3. *a) If a split destroys an i-cap and an i-cup for some $i \in [m]$ and these are the leftmost i-cap and i-cup, then the split is i-feasible.*

b) If a split destroys an i-cap (respectively i-cup) for some row $i \in [m]$ with split interval I and all the other i-caps and i-cups have split intervals that are disjoint from I, then the split is i-feasible.

c) Let us consider a trunk $(I_1, \ldots, I_{\tilde{k}})$ in some row i. A split $s_{B_{ij}, B_{i,j+1}}$ with $j \in I_1 \setminus I_2$ is i-infeasible iff $\tilde{k}$ is even.

We propose the following

Splitting procedure: Iteratively insert feasible splits until no more obstacles exist in the whole matrix.

Obviously, at the end there are exactly $s_i(A)$ splits and $c_i^{TG}(A)$ connected regions of ones in each row $i \in [m]$.

The only thing we still have to prove is that the choice of a feasible split in the splitting procedure is always possible.

Lemma 4. *Let the binary matrix A and a set of feasible splits $\mathcal{SP}$ be given such that there is still a connected region of ones that does not satisfy the consecutive ones property. Then there exists another feasible split.*

Proof. As there is a connected region of ones that does not satisfy the consecutive ones property, there exists an i-cap or an i-cup for some row $i \in [m]$. We consider the i-obstacle with the leftmost critical zero and under this circumstance minimal value of i.

We can again w.l.o.g. assume that this is a subset of ones of the form of an i-cup, because the case of an i-cap is similar. Let the leftmost split interval in row i be $[j, j'-1]$ and no split of type $s_{B_{ik}, B_{i,k+1}}$ with $k \in [j, j'-1]$ is already in $\mathcal{SP}$. Let i' be the last row below row i, for which this is also an i'-cup. Possibly, $i' = i$. Since $[j, j'-1]$ is the split interval of the leftmost i-cup we know that all the trunks which contain $[j, j'-1]$ start in that split interval. We have the following situation where at least one of the $*$-positions is a 0:

$$\begin{array}{c c} & \begin{array}{ccccc} j & & & & j' \end{array} \\ \begin{array}{c} i-1 \\ i \\ \\ i' \\ i'+1 \end{array} & \begin{pmatrix} 1 & 0 & \ldots & 0 & 1 \\ 1 & 1 & \ldots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \ldots & 1 & 1 \\ * & * & \ldots & * & * \end{pmatrix} \end{array}$$

To produce connected regions satisfying the consecutive ones property, we have to show that one of the splits $s_{B_{i,k}, B_{i,k+1}}$ for $k \in [j, j'-1]$ is feasible. Each of these splits affects at least the rows $k \in [i, i']$ for which there is the k-cup with split interval $[j, j'-1]$. We distinguish different cases:

Case 1: For all $k \in [i, i']$, there is no k-cap with split interval $[\ell, \ell'-1]$ such that $[j, j'-1] \cap [\ell, \ell'-1] \neq \emptyset$. As there is at least one zero in row $i'+1$ in $[j, j']$, there is a split that only affects rows $k \in [i, i']$ for which there is the k-cup with split interval $[j, j'-1]$.

Using Lemma 3 b) we obtain the feasibility of this split.

Case 2: There is an interval $[k_1, k_2]$ with $i \le k_1 \le i'$ such that there is a k-cap with split interval $[\ell, \ell'-1]$ such that $[j, j'-1] \cap [\ell, \ell'-1] \neq \emptyset$ for all $k \in [k_1, k_2]$. Obviously, $k_2 < i'$ is not possible because of Lemma 2 b). If $k_2 = i'$ every split in $[j, j'-1] \cap [l, l'-1]$ is feasible using Lemma 3 a) and b). Let us therefore consider the case $k_2 > i'$:

$$\begin{array}{l c}
 & \begin{array}{ccccccc} j & \ell & & & j' & & \ell' \end{array} \\
\begin{array}{l} i-1 \\ i = i' = k_1 \\ k_2 \\ \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ * & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{array}$$

Let us assume that the splits in $[j, j'-1] \cap [\ell, \ell'-1]$ are k-infeasible for some row $k \in [i'+1, k_2]$ (if not, there is a feasible split). That means, in row k it is not allowed to split only the k-cap in $[\ell, \ell'-1]$, because it has to be cut together with a k-cup on the right. Lemma 3 c) tells us that the trunk of split intervals in row k starting with $[\ell, \ell'-1]$ ends with a split interval corresponding to a k-cup (because the total number of vertices in that trunk must be even). Thus the trunk of split intervals in row k is of the form $[\ell, \ell'-1] = J_1, I_1, J_2, I_2, \dots, J_t, I_t$, where $J_1, \dots, J_t$ are k-caps and the $I_1, \dots, I_t$ are k-cups. Now we use Lemma 2 several times: Because row k and the rows in $[k_1, i']$ share J_1, Lemma 2 tells us that they also share I_1. Now there are two parts: For the rows in $h \in [k_1, i']$ which do not share J_2, the trunk of these rows starts and ends with a cup and thus every split in $[j, j'-1] \setminus [\ell, \ell'-1]$ is h-feasible. And for the rows in $h \in [k_1, i']$ which share J_2, we use Lemma 2 again and we obtain, that they share I_2 and so on. Thus, for all $h \in [k_1, i']$, we either find an h-feasible split in $[j, j'-1] \setminus [\ell, \ell'-1]$ (*) or row h shares all the split intervals with row k (**). Furthermore, the trunks of split intervals in rows $h \in [k_1, i']$ cannot be longer than $([j, j'-1], J_1, I_1, J_2, I_2, \dots, J_t, I_t)$, as an i'-cap would have to follow that, again using Lemma 2 would also be a k-cap, a contradiction. Thus, the trunk for the rows $h \in [k_1, i']$ is exactly $([j, j'-1], J_1, I_1, J_2, I_2, \dots, J_t, I_t)$ in case (**) and the number of split intervals is again odd. Again, every split in $[j, j'-1] \setminus [\ell, \ell'-1]$ is h-feasible. All in all, every split in $[j, j'-1] \setminus [\ell, \ell'-1]$ is feasible, as it is h-feasible for all the split rows h.

□

The result of our splitting procedure is the following: We have inserted a number of feasible splits, until no more feasible splits are possible. Afterwards each row i is split exactly $s_i(A)$ times and all the connected regions of ones in the matrix have the consecutive ones property. We have $c_i^{TG}(A)$ connected regions of ones intersecting with row i for all $i \in [m]$. The splitting procedure takes time $O(m^2 n^2)$. At first, in each row $i \in [m]$ and for each block B_{ij} we need at most mn operations to check, if a split after this block is necessary. Thus, it takes time $O(m^2 n^2)$ to find all split intervals for all rows. With these split intervals it takes $O(n)$ to find $s_i(A)$ in each row i. Afterwards, checking that a split is i-feasible can be done in time $O(n)$ by computing the minimal number of splits for the left part and for the right part.

We will now define a step of the segmentation procedure, that finds for given A a TG-segment S such that $A - S$ is nonnegative and $c^{TG}(A-S) = c^{TG}(A) - 1$. Let us assume that we have already obtained the set $\mathcal{SP}$ of splits from the splitting procedure, i.e. we have a number of connected regions of ones with consecutive ones property, whose union is the set of ones in A. We call a row $i \in [m]$ *critical* if $c_i^{TG}(A) = c^{TG}(A)$. For $i \in [m]$ let $\boldsymbol{s}_i$ denote the i-th row of S.

Algorithm 1 Segmentation

Input: Matrix A with splits
  $s_{ij} = 0$ for all $(i, j) \in [m] \times [n]$
  for $i = 1$ to m **do**
    if i is critical and $\boldsymbol{s}_i = \mathbf{0}$ **then**
      Choose a connected region of ones that intersects row i but no row $k < i$ with $\boldsymbol{s}_k \neq \mathbf{0}$.
      Add this connected region of ones to S.
    end if
  end for
  $A = A - S$
Output: Matrix A

We prove in Lemma 5 that we can always find such a region for each critical row, which is still empty in S. Because the segmentation procedure selects only connected regions of ones from A it obviously follows that $A - S$ is nonnegative. Moreover, because all critical rows i satisfy $\boldsymbol{s}_i \neq \mathbf{0}$ at the end of the for-loop, we also have that $c^{TG}(A-S) = c^{TG}(A) - 1$. Hence the segmentation procedure will lead us to a segmentation of A which uses $c^{TG}(A)$ TG-segments, when we iterate it until $A = \mathbf{0}$. The only thing we still have to check is the fact that for each critical row i such that still $\boldsymbol{s}_i = \mathbf{0}$ in the for-loop we can always find a connected

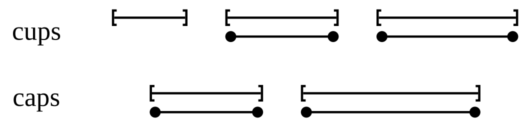


Fig. 9. Trunks of split intervals. The intervals with the circles at the end are the k-obstacles, the others those of the rows in $[k_1, i']$.

region of ones that does not intersect a non-empty row in the current segment S (see Figure 10) and that can be added to S.

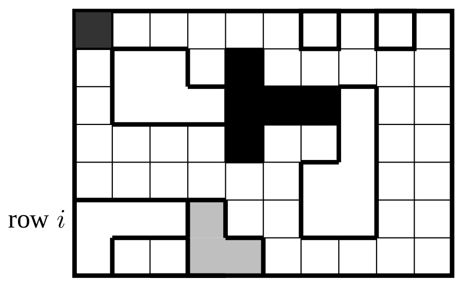


Fig. 10. A matrix A with its connected regions of ones with respect to the splits from the splitting procedure. The black areas form the current segment S in the for-loop of the segmentation algorithm. The first critical row which is still empty in S is row $i = 6$. The grey area intersects with row $i = 6$ but does not intersect with non-empty rows of S. So, we can choose this connected region of ones to complete S.

Lemma 5. *Let the matrix A and its splits be given and let a number of connected regions of ones be already chosen that form a current segment S. Let $i \in [m]$ be a critical row with $\boldsymbol{s}_i = \mathbf{0}$. Let $\boldsymbol{s}_k = \mathbf{0}$ for all $k \geq i$. Then there exists a connected region of ones in A that intersects with row i but with no row $k < i$ with $\boldsymbol{s}_k \neq \mathbf{0}$.*

Proof. Let us assume all the connected regions of ones that intersect with row i cannot be added to S because they intersect with some row $k < i$. Let k^* be the largest index of a nonzero row in S. Because of the connectedness, each connected regions of ones from row i intersects with row k^*. As there are $c^{TG}(A)$ connected regions of ones intersecting with row i, there are at least $c^{TG}(A) + 1$ regions of ones intersecting with row k^* in contradiction to i being a critical row. Thus, the assumption was wrong and we find a region of ones in row i that can be added to S. □

Note, that after subtracting a segment S from A, we can use the algorithm above again, but it is not necessary to compute the splitting procedure for the updated matrix A. We can just use the old partition where some of the splits have become useless.

Theorem 1. *The minimal beam-on time of a segmentation of A into TG-segments is $c^{TG}(A)$.*

Proof. It is obvious that we need at least $c^{TG}(A)$ TG-segments to decompose A because there is some row i^* whose boxes can only be decomposed by at least $c^{TG}_{i^*}(A) = c^{TG}(A)$ segments. After the splitting procedure, we find at most $c^{TG}(A)$ regions of ones in each row $i \in [m]$ and eliminating one of them always corresponds to decreasing the TG row complexity of row i by 1. Obviously, Algorithm 1 finds a TG-segment that decreases the TG row complexity by 1 in all the critical rows (and maybe also in some other rows). The statement then follows by induction. □

Corollary 2. *The optimal decomposition of a binary input matrix into TG-segments can be found in polynomial time.*

Proof. The splitting procedure takes time $O(m^2n^2)$ and produces less than mn connected regions of ones. Checking if a connected region of ones should be added in the segmentation procedure also takes time $O(mn)$. Thus, the whole decomposition can be done in time $O(m^2n^2)$. □

We close this section with an
Example 1. *We discuss our whole approach using the example matrix*

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1| & 1 & 1 & 1| & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1| & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \textit{and}$$

$$S_1 = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&1&0&0&0&0&0 \end{pmatrix}$$

from Figure 10 with the splits indicated by vertical bars. The first segment S_1 is determined after computing the TG-row-complexities

i	*1*	*2*	*3*	*4*	*5*	*6*	*7*
$c_i(a)$	*3*	*2*	*1*	*2*	*1*	*2*	*2*
$s_i(A)$	*0*	*0*	*2*	*0*	*0*	*1*	*0*
$c_i^{TG}(A)$	*3*	*2*	*3*	*2*	*1*	*3*	*2*

deducing $c^{TG}(A) = 3$, applying the splitting procedure and the first step of the segmentation. After inserting feasible splits, the connected regions of ones are according to Figure 10. In the first step of the segmentation, the critical rows are the rows 1, 3 and 6. After removing S_1, the critical rows are the rows 1, 3 and 6 again, where the row complexity is 2 now. The next two steps of the segmentation procedure then might produce

$$S_2 = \begin{pmatrix} 0&0&0&0&0&0&1&0&0&0 \\ 0&1&1&0&0&0&0&0&0&0 \\ 0&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0 \end{pmatrix} \quad \textit{and}$$

$$S_3 = \begin{pmatrix} 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

which finally yields an optimal TG-decomposition with three TG-segments.

3. Relation to colorings of perfect graphs

We show that finding the optimal beam-on time of a TG-segmentation of a binary matrix is equivalent to computing the chromatic number in a perfect graph. This gives an alternative proof that the problem can be solved in polynomial time.

The *chromatic number* of a graph $G = (V, E)$ is the minimal number of colors we need to color the vertices of G such that no two adjacent vertices have the same color. This number is denoted by $\chi(G)$. A *clique* in G is a subset of vertices, such that each two of them are adjacent. The size of a largest clique in G is denoted by $\omega(G)$.

A *perfect graph* is a graph G in which the chromatic number of every induced subgraph equals the size of the largest clique of that subgraph, i.e. G is perfect if for every induced subgraph G' of G we have $\chi(G') = \omega(G')$.

Let the binary matrix A be given. We define a graph $G_A = (V_A, E_A)$ as follows: The set of vertices V_A is the set of boxes of A. The set of edges E_A is the set of pairs of boxes (B, B') such that B and B' are not allowed to be in the same segment of a TG-segmentation of A. That means, two boxes $B = [i_1, i_2] \times \{j\}$ and $B' = [i'_1, i'_2] \times \{j'\}$ are adjacent, if $[i_1, i_2] \cap [i'_1, i'_2] \neq \emptyset$ and if

- either there is an entry $a_{i,j''} = 0$ for some $i \in [i_1, i_2] \cap [i'_1, i'_2]$ and $j'' \in [j + 1, j' - 1]$ or
- there is some $j'' \in [j + 1, j' - 1]$ such that for all rows $i \in [i_1, i_2] \cap [i'_1, i'_2]$ there is an i-obstacle with split interval $[j, j'']$ or $[j'', j']$.

For example, in Figure 11 box 4 and 10 from the left would be adjacent, because the boxes 7 to 10 form an i-cap. Note that, if two boxes belong to the same connected region of ones resulting from the splitting procedure, they are non-adjacent in the graph. This graph is called the *TG-graph* of A.

Theorem 3. $c^{TG}(A) = \omega(G_A)$

Proof. $c^{TG}(A) \geq \omega(G_A)$ is easy to see, as each box in a maximal clique of G_A needs its own segment. Let now i be a row with $c_i^{TG}(A) = c^{TG}(A)$, i.e. after applying the splitting procedure the boxes of row i decompose into $c^{TG}(A)$ many connected regions of ones such that each two of them have to be irradiated separately. As we show now, it is possible to choose one box from each region of ones, such that each two of the chosen boxes are not allowed to be put into the same segment of a TG-segmentation of A. The choice of the boxes from each region of ones is illustrated in Figure 11 and can be realized as follows:

- We go through the boxes $B_{i1}, \ldots, B_{in}$ of row i from left to right. W.l.o.g. we only have to discuss the choice of boxes within a sequence $B_{ij}, \ldots, B_{ij'}$ of nonempty boxes (i.e. $a_{ik} \neq 0$ for $j \leq k \leq j'$), because if there is a zero in row i between two boxes they are adjacent in the graph anyway. Let us therefore consider such a sequence of consecutive connected regions of ones resulting from the splitting procedure such that all the involved boxes are nonempty.

- We start with the leftmost region of ones and choose its leftmost box for the clique. We then always go to the next region of ones on the right and choose the leftmost box in this region for the clique that is adjacent in the graph G_A with the previously chosen box.
- This indeed gives a clique, because if we assume two chosen boxes $B = [i_1, i_2] \times \{j\}$ and $B' = [i'_1, i'_2] \times \{j'\}$ with $i_1 \leq i \leq i_2$ and $i'_1 \leq i \leq i'_2$ are non-adjacent, this would mean that the boxes $B_{ij}, B_{i,j+1}, \ldots, B_{i,j'-1}, B_{i,j'}$ satisfy the consecutive ones property, a contradiction to the choice of boxes described above.

Using this procedure, the chosen $c^{TG}(A)$ many boxes form a clique in G_A and we have $c^{TG}(A) \leq \omega(G_A)$. All in all, we have $c^{TG}(A) = \omega(G_A)$.

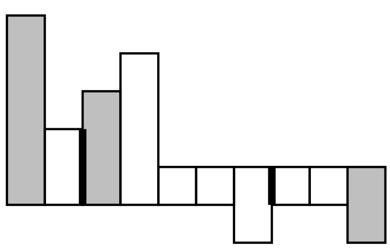


Fig. 11. The bold lines indicate the possible split positions and the grey boxes are chosen to form a maximal clique.

□

Theorem 4. $c^{TG}(A) = \chi(G_A)$

Proof. By definition, the chromatic number is the minimal number of stable sets we need to decompose a graph, as each color has to be assigned to a stable set of vertices. Obviously, $\chi(G_A) \leq c^{TG}(A)$, as each segment exactly corresponds to a stable set in G_A and therefore an optimal segmentation yields a coloring with $c^{TG}(A)$ many colors.

Furthermore, we have $\omega(G_A) \leq \chi(G_A)$, as this holds for every graph. Together with Theorem 3, we have

$$\omega(G_A) \leq \chi(G_A) \leq c^{TG}(A) = \omega(G_A)$$

and thus $c^{TG}(A) = \chi(G_A)$. □

Theorems 3 and 4 together give $\chi(G_A) = \omega(G_A)$ for the TG-graph G_A of A. If we consider induced subgraphs of G_A, the boxes that correspond to the chosen subset of vertices form a binary matrix, that we call A' from now on. Note that the induced subgraph of G_A that has the boxes of A' as vertices, denoted by H, is not necessarily $G_{A'}$. It can happen, that two boxes B and B' of A' are adjacent in $G_{A'}$, but not adjacent in the induced subgraph H, because other boxes of A, that do not belong to A', allowed putting B and B' into the same segment. This is illustrated in Figure 12.

Theorem 5. *For every induced subgraph H of G_A there exists a graph $G_{A''}$ that is the TG-graph of a binary matrix A'' with $\chi(H) = \chi(G_{A''})$ and $\omega(H) = \omega(G_{A''})$.*

Proof. Let H be an arbitrary induced subgraph of G_A such that the boxes of the induced vertex set form a binary matrix A'. Let A'' be the matrix that has the same boxes as A' and some extra boxes defined as follows: Whenever we have two boxes $B = [i_1, i_2] \times \{j\}$ and $B' = [k_1, k_2] \times \{\ell\}$ in A' with $\ell > j$ such that B and B' are not adjacent in G_A and such that there are only zeros in $([i_1, i_2] \cap [k_1, k_2]) \times [j+1, \ell-1]$ in A', then we add the intermediate boxes $([i_1, i_2] \cap [k_1, k_2]) \times \{t\}$ for all $t \in [j+1, \ell-1]$ to A''. For example, the three white boxes in Figure 12 on the left are the intermediate boxes of B and B'. Therefore, B and B' can be put into the same segment of a TG-segmentation of A''. Let $G_{A''}$ be the TG-graph corresponding to the matrix A''. $G_{A''}$ has more vertices than H and some extra edges that are incident with the new vertices. Note that two boxes of A' that are adjacent in H are also adjacent in $G_{A''}$, because they still cannot be in the same segment. Similarly, two boxes of A' that are non-adjacent in H are non-adjacent in $G_{A''}$, as we inserted the intermediate boxes. Thus, H is an induced subgraph of $G_{A''}$ and $\chi(H) \leq \chi(G_{A''})$ and $\omega(H) \leq \omega(G_{A''})$ is immediately obvious.

Let now B and B' be two such boxes of A' such that we inserted the intermediate boxes between them in A''. It is easy to verify that the intermediate boxes are not adjacent to B and B' in $G_{A''}$ and also not adjacent to all non-neighbors of B and B' in $G_{A''}$ (and also in H, as these non-neighbors are the same). This is the case, because the intermediate boxes can be put into the same segment with all boxes that can be put into the same segment with either B or B'. Thus, if we have an optimal decomposition of H into stable sets, we can put all the intermediate boxes into the stable

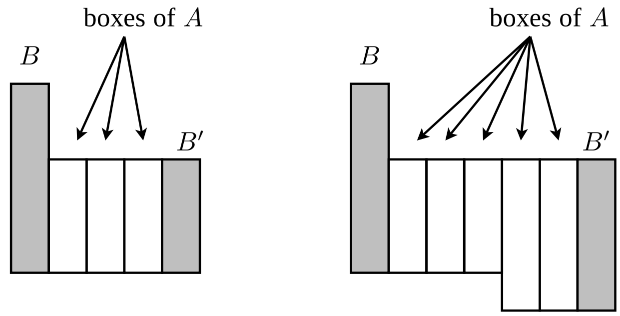


Fig. 12. The white boxes are boxes of A that are not present in A'. These are two examples where B and B' are not adjacent in the induced subgraph, because there were other boxes of A that made their combination possible.

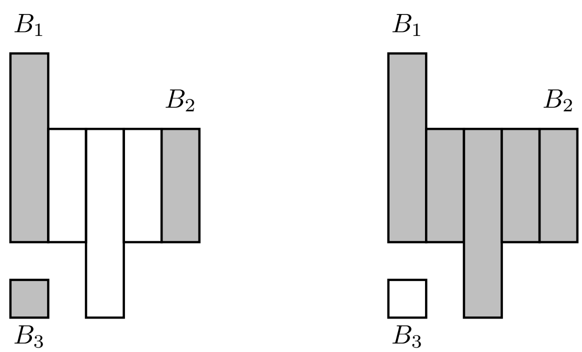


Fig. 13. The same boxes with two different stable set decompositions (the grey boxes and the white boxes each form a stable set). The decomposition on the right gives a TG-segmentation, but the decomposition on the left does not.

set containing B or the stable set containing B' and get stable sets in $G_{A''}$. Doing this for all pairs B and B' where we have intermediate boxes yields a stable set decomposition of $G_{A''}$ with $\chi(H)$ many stable sets. Thus, $\chi(H) \geq \chi(G_{A''})$.

Let us now consider a largest clique in $G_{A''}$. If this clique contains no intermediate boxes, this is a clique in H. If it contains intermediate boxes, we do the following substitution: For every boxes B and B' of A' where we have intermediate boxes in between, only either B or B' or one of the intermediate boxes can be in the clique, as they are all non-adjacent. If an intermediate box is contained in the maximal clique, we delete the intermediate box and put either B or B' into the maximal clique. This is possible, because every box that cannot be in the same segment with the intermediate box also cannot be in the same segment with B and B' and thus all neighbors of the intermediate box are also neighbors of B and B'. The question arises if we might need a box B twice for substitution because there are intermediate boxes left and right from B. But this cannot happen because in a sequence B'',B, B' with intermediate boxes between B'' and B and between B and B', all intermediate boxes (left and right from B) and B are non-adjacent (as they can be in the same segment). Therefore, there never can be two intermediate boxes of the sequence B'', B, B' with intermediate boxes between B'' and B and between B and B' in a maximal clique. After the substitution procedure we have found a clique of the same cardinality containing only boxes from A'. These boxes form a clique in H and thus we get $\omega(H) \geq \omega(G_{A''})$. This concludes the proof. □

Using Theorems 3, 4 and 5 we get the following

Corollary 6. *The graph G_A is a perfect graph with* $\chi(G_A) = c^{TG}(A)$.

As the coloring problem in perfect graphs can be solved in polynomial time, the beam-on time problem for TG-segmentations is also polynomial. We remark that, although the chromatic number of G_A gives the optimal beam-on time of a TG-segmentation of A, not all optimal colorings of G_A yield a TG-segmentation of A. For example, if the stable set decomposition is like in Figure 13 on the left, we get no feasible TG-segmentation, as there are two boxes in a stable set that only form a segment if the intermediate boxes are in the same stable set (like in the decomposition on the right).

Thus, we should ask ourselves how we can modify an optimal stable set decomposition of G_A in such a way that each stable set really represents a segment. But from an algorithmic point of view, this question is not interesting, as the algorithms that solve the coloring problem in a perfect graph are slower than the one we have presented in Section 2.

4. Conclusion

We have proved that for binary input matrices the problem of finding a decomposition which minimizes the beam-on time or the cardinality under the tongue-and-groove constraint is polynomial. Obviously it remains the question whether the beam-on time problem under the tongue-and-groove constraint is still polynomial for integer input matrices. Up to now, we do not see that the tools we developed in this paper can be generalized to provide a result for the integer case.

5. Acknowledgments

We thank Konrad Engel and Thomas Kalinowski from the University of Rostock and Samuel Fiorini from the Université Libre de Bruxelles for their constructive remarks during the redaction of this paper.

References

[1] R.K. Ahuja and H.W. Hamacher. A network flow algorithm to minimize beam-on time for unconstrained multileaf collimator problems in cancer radiation therapy. *Networks*, 45(1):36–41, 2005.

[2] 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.

[3] 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.

[4] 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.

[5] R. Burkart. Open Problem Session, Oberwolfach Conference on Combinatorial Optimization, November 24-29, 2002.

[6] M.J. Collins, D. Kempe, J. Saia, and M. Young. Nonnegative integral subset representations of integer sets. *Inform. Process. Lett.*, 101(3):129–133, 2007.

[7] M. Ehrgott, Ç. Güler, H.W. Hamacher, and L. Shao. Mathematical optimization in intensity modulated radiation therapy. *4OR*, 6(3):199–262, 2008.

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

[9] K. Engel and T. Gauer. A dose optimization method for electron radiotherapy using randomized aperture beams. *Phys. Med. Biol.*, 54(17):5253–5270, 2009.

[10] C. Engelbeen and S. Fiorini. Constrained decompositions of integer matrices and their applications to intensity modulated radiation therapy. *Networks*, 2009. DOI 10.1002/net.20324.

[11] C. Engelbeen, S. Fiorini, and A. Kiesel. A closest vector problem arising in radiation therapy planning. *Journal of Combinatorial Optimization*, DOI 10.1007/s10878-010-9308-8, 2010.

[12] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57 of *Annals of Discrete Mathematics*. Elsevier Science B.V., Amsterdam, 2004.

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

[14] T. Kalinowski. Multileaf collimator shape matrix decomposition. In G.J. Lim and E.K. Lee, editors, *Optimization in Medicine and Biology*, pages 253–286. Auerbach Publishers Inc., 2008.

[15] T. Kalinowski. Reducing the tongue-and-groove underdosage in MLC shape matrix decomposition. *Algorithmic Operations Research*, 3(2), 2008.

[16] 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.

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

[18] S. Kamath, S. Sahni, S. Ranka, J. Li, and J. Palta. A comparison of step–and–shoot leaf sequencing algorithms that eliminate tongue–and–groove effects. *Phys. Med. Biol.*, 49:3137–3143, 2004.

[19] A. Kiesel and T. Gauer. Approximated segmentation considering technical and dosimetric constraints in intensity modulated radiation therapy with electrons. arXiv:1005.0898, 2010.

[20] 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.

[21] 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.

[22] 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.

[23] 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 25-12-2009; revised 19-5-2010; accepted 5-6-2010