Optimal multileaf collimator field segmentation

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)
der Mathemathisch–Naturwissenschaftlichen Fakultät
der Universität Rostock

vorgelegt von
Thomas Kalinowski, geb. am 27.11. 1980 in Pasewalk
aus Rostock

Rostock, 1. September 2004

Dekan:	Prof. Dr. Udo Kragl
Gutachter:	Prof. Dr. K. Engel (Rostock)
Gutachter:	Prof. Dr. H. Hamacher (Kaiserslautern)
Gutachter:	Prof. Dr. R. Ahlswede (Bielefeld)
Tag der öffentlichen Verteidigung:	28.01.2005

ACKNOWLEDGEMENT

It is a great pleasure for me to thank Prof. Konrad Engel for his reliable advice concerning all questions arising in connection with my dissertation, and especially for his patience with several versions of this work. I also thank my family and friends for always supporting me. Special thanks to Markus Stoy for many useful LaTeX hints.

CONTENTS

1. INTRODUCTION

In cancer treatment high energetic radiation is used to destroy the tumor. To achieve this goal the irradiation process must be planned in such a way that the tumor (target volume) receives a sufficiently high dose while the organs close to it (organs at risk) are not damaged. In clinical practice the radiation is delivered by a linear accelerator and the beam head is part of a gantry that can be rotated about the treatment couch (see Figure 1.1). In order to design

Fig. 1.1: A treatment couch with gantry and beam head.

a treatment plan a set of gantry angles and corresponding beam intensities have to be determined. See [13, 21, 22] for reviews of different approaches to this problem. Because of technological developments recently intensity modulated radiotherapy (IMRT) became an important method to improve the quality of the treatment. Here instead of one rectangular homogeneous field a number of differently shaped fields are superimposed, and so an intensity modulated field is produced. The shapes of the fields are determined by a multileaf collimator (MLC). A multileaf collimator consists of two opposing banks of metal leaves (see Figure 1.2). The leaves can be moved independent of each other and each leaf configuration corresponds to a radiation field of a particular shape that is realized by inserting the MLC between the radiation source and the patient. Now the realization of an intensity modulated field

Fig. 1.2: The leaf pairs of a multileaf collimator.

using an MLC can be modeled as follows. The beam head is discretized into bixels and the modulated field is described by giving the desired intensity for each bixel. So the intensity map can be considered as an $m \times n-$matrix, where each row corresponds to a leaf pair. There are two essentially different ways to generate intensity modulated fields with multileaf collimators: in the static mode (step–and–shoot) [6–8, 10, 11, 16, 20, 23, 28] the beam is switched off when the leaves are moving while in the dynamic mode [7, 9, 17, 19, 24–27] the beam is switched on during the whole treatment and the modulation is achieved by varying the speed of the leaf motion. Two important criteria for the quality of a treatment plan are the total irradiation time and the total treatment time. The total irradiation time should be small since there is always a small amount of radiation transmitted through the leaves, and if the used model ignores this leaf transmission the error increases with the total irradiation time. A small total treatment time is desirable for efficiency reasons. In the dynamic mode the two criteria coincide. So here the problem is to determine a velocity function for each leaf such that the given intensity is realized in the shortest possible time. In the static mode the whole treatment consists of the irradiation and the intervals in between when the leaves are moved. Thus we have two parameters which influence the total treatment time: the irradiation time and the number of homogeneous fields that are needed. How these parameters have to be weighted depends on the used technology: the longer the time intervals between the different fields are, the more the reduction of the number of fields becomes important. The lengths of these time intervals are influenced by the leaf velocity and by the so called verification and record overhead, which is the time necessary to check the correct positions of the leaves. In a more realistic model one should also take the shapes of the fields into account, because clearly the necessary leaf travel time between two fields depends on the shapes of these fields [4, 23]. The dynamic mode has the advantage of a smaller total treatment time, but the static mode involves no leaf movement with radiation on and so the verification of the correct realization of the treatment plan is easier which makes the method less sensitive to malfunctions of the technology.

There are additional machine–dependent restrictions which have to be considered when determining the leaf positions:

Interleaf collision constraint (ICC): In some widely used MLCs it is forbidden that opposite leaves of adjacent rows overlap, because otherwise these leaves collide. So leaf positions as illustrated in Figure 1.3 (where the shading indicates the area that is covered by the leaves) are not allowed.

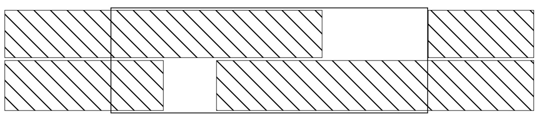


Fig. 1.3: Leaf position that is excluded by the ICC.

Tongue and groove constraint: In order to reduce leakage radiation between adjacent leaves the commercially available MLCs use a tongue–and–groove (or similar) design with the effect that there is a small overlap of the regions that are covered by adjacent leaves. This is illustrated in Figure 1.4 which shows a cut through the leaf bank perpendicular to the direction of leaf motion.

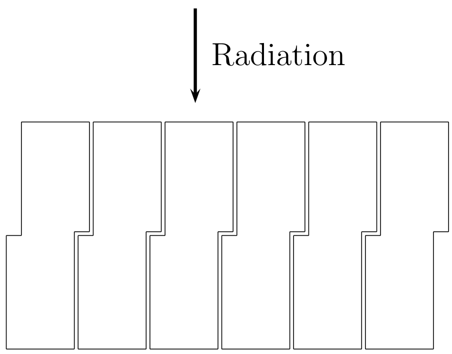
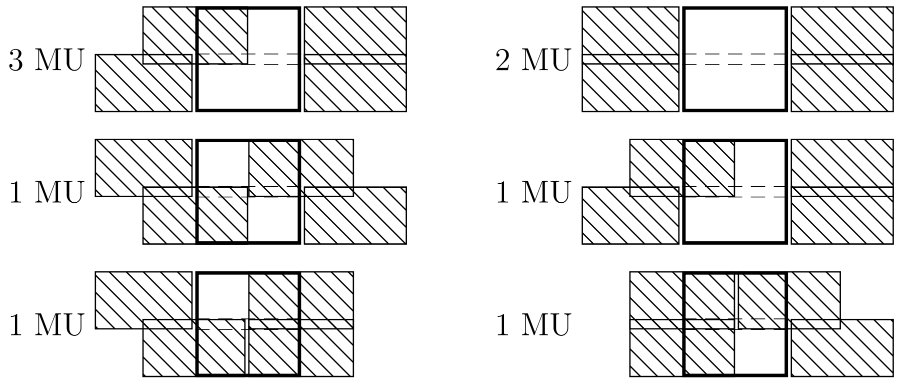


Fig. 1.4: The principle of the tongue–and–groove design.

Consider two bixels x and y that are adjacent along a column and two homogeneous fields, where in the first field x is irradiated and y is covered and in the second field y is irradiated and x is covered. Then in the composition of these fields along the border of x and y there is a narrow strip (the overlap of the regions that are covered by the leaves in the rows of x and y, respectively) that receives no radiation at all. Figure 1.5 illustrates this for the intensity map $\left(\begin{smallmatrix}2&3\\3&4\end{smallmatrix}\right)$.



a) The overlap of bixels $(1,1)$ and $(2,1)$ receives no radiation because of the tongue and groove effect.

b) The overlaps of bixels that are adjacent along a column receive the smaller one of the relevant doses.

Fig. 1.5: Two different realizations of the same intensity map.

To avoid this effect one may require that two bixels that are adjacent along a column are irradiated simultaneously for the time the lower of the two doses is delivered. Then the border region receives this lower dose. If this is the case for all the relevant pairs of adjacent bixels the treatment plan is said to satisfy the tongue and groove constraint.

In this work we consider the problem of constructing segmentations, taking into account the interleaf collision constraint, while we neglect the tongue and groove constraint. Describing the possible leaf positions of the MLC by certain $(0,1)$–matrices, called segments, this amounts to the search for a realization of the given intensity matrix A with a small total number of monitor units (TNMU) and a small number of segments (NS). In general it is not possible to minimize both parameters simultaneously [15]. For the case of an MLC without ICC there are several segmentation algorithms [6–8, 11, 20, 23, 28], some of them providing the optimal TNMU but a large NS, others reducing the NS heuristically at the price of an increased TNMU. The exact minimization of the NS is NP–hard already for intensity matrices consisting of just one row [2, 15]. In principle both, TNMU and NS, can be optimized by integer programming and this method can be adapted to additional restrictions like ICC [18]. But clearly this is applicable only for small problem sizes. Another approach is the reformulation of the segmentation problem in a network flow setting. In [4] this is done for MLC–segmentation with ICC, yielding a network flow algorithm for the TNMU–optimal segmentation. In [2] this approach is developed further and a heuristic for the reduction of the NS is added. The method of [23] yielding TNMU–optimal segmentations without ICC is modified in [16] such that it takes into account the ICC (and an even more general condition). In [10] there is presented an efficient segmentation algorithm yielding the optimal TNMU and a very small NS for the segmentation problem without ICC. See [15] for a survey and a comparison of the different segmentation algorithms. Engel's algorithm [10] for the segmentation without ICC is derived from an explicit formula for the smallest possible TNMU. Evaluating this formula is equivalent to solving a longest path problem in a properly constructed layered digraph. Theorem 1, one of the main results of this thesis, is a generalization of that construction such that the longest path problem in the new digraph corresponds to the evaluation of the minimal TNMU for a segmentation with ICC. Chapter 2 is devoted to the proof of this theorem which consists of two parts: using a duality argument we establish that the maximal weight of a path in the digraph to be described below is a lower bound for the number of monitor units in a segmentation, and then we describe a method to achieve this bound. In Chapter 3 we develop a heuristic method to construct a segmentation with minimal number of monitor units and a small number of segments. In Chapters 4 and 5 we discuss some results for MLC segmentation without ICC. In Chapter 4 we show that a segmentation with in first instance minimal TNMU and in second instance minimal NS can be determined in polynomial time if the entries of the intensity matrix are bounded by a constant. Finally, in Chapter 5 we suggest two more flexible ways of using the MLC, thus increasing the number of possible shapes for the homogeneous fields. In 5.1 we study the situation when it is allowed to rotate the MLC about 90° between the delivery of two segments and in 5.2 we consider a pair of two MLCs with orthogonal directions of leaf motion.

2. AN ALGORITHM FOR TNMU–OPTIMAL FIELD SEGMENTATION WITH INTERLEAF COLLISION CONSTRAINT

2.1 A Linear Programming formulation

Throughout m and n are positive integers and for positive integers k we use the notation $[k] := \{1, 2, \dots, k\}$. Let $A = (a_{i,j})_{\substack{1\le i\le m\\ 1\le j\le n}}$ be an $m\times n$-matrix with nonnegative integer entries. In addition we put $a_{i,0} = a_{i,n+1} = 0$ $(i \in [m])$. A segment is a matrix that corresponds to a position of an MLC with interleaf collision constraint. This is made precise in the following definition.

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

$$l_i \le r_i + 1 \qquad (i \in [m]), \tag{2.1}$$

$$s_{i,j} = \begin{cases} 1 & \text{if } l_i \le j \le r_i \\ 0 & \text{otherwise} \end{cases} \qquad (i \in [m],\ j \in [n]), \tag{2.2}$$

$$\text{ICC:} \quad l_i \le r_{i+1} + 1,\ r_i \ge l_{i+1} - 1 \qquad (i \in [m-1]). \tag{2.3}$$

The interpretation is that $l_i - 1$ and $r_i + 1$ are the positions of the i–th left and right leaf, respectively. A *segmentation* of A is a representation of A as a sum of segments, that is $A = \sum_{i=1}^{k} u_i S_i$ with segments S_i $(i = 1, 2, \dots, k)$ and positive integers u_i $(i = 1, 2, \dots, k)$. The TNMU of this segmentation is $\sum_{i=1}^{k} u_i$ and our goal is to find a segmentation of A with minimal TNMU.

Example 1. A segmentation with 10 MU for a benchmark matrix from [18] is

$$\begin{aligned} \begin{pmatrix} 4&5&0&1&4&5\\ 2&4&1&3&1&4\\ 2&3&2&1&2&4\\ 5&3&3&2&5&3 \end{pmatrix} &= 3\begin{pmatrix} 0&0&0&0&1&1\\ 0&0&0&0&0&1\\ 0&0&0&0&0&1\\ 0&0&0&0&1&1 \end{pmatrix} + 3\begin{pmatrix} 1&1&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&0&0&0&0\\ 1&0&0&0&0&0 \end{pmatrix} + 1\begin{pmatrix} 0&0&0&0&0&1\\ 0&0&0&1&1&1\\ 0&0&0&0&1&0\\ 0&1&1&1&1&0 \end{pmatrix} \\ &+ 1\begin{pmatrix} 0&0&0&1&1&1\\ 0&0&0&1&0&0\\ 0&1&1&0&0&0\\ 1&1&1&0&0&0 \end{pmatrix} + 1\begin{pmatrix} 0&1&0&0&0&0\\ 1&0&0&0&0&0\\ 1&1&0&0&0&0\\ 1&1&1&1&1&0 \end{pmatrix} + 1\begin{pmatrix} 1&1&0&0&0&0\\ 1&1&1&1&0&0\\ 1&1&1&1&1&1\\ 0&0&0&0&0&0 \end{pmatrix}. \end{aligned} \tag{2.4}$$

By $\mathcal{F}$ we denote the family of subsets of $V := [m] \times [n]$ that correspond to segments, precisely

$$\mathcal{F} = \{T \subseteq V : \text{ There exists a segment } S \text{ with } (i,j) \in T \iff s_{i,j} = 1\}.$$

With a segmentation $A = \sum_{i=1}^{k} u_i S_i$ we can associate a function $f : \mathcal{F} \to \mathbb{N}$: for $1 \leq i \leq k$ we put $f(T) = u_i$ for the $T \subseteq V$ corresponding to the segment S_i, and for the remaining T we put $f(T) = 0$. Now the LP-relaxation of the segmentation problem is:

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

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

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

To solve the segmentation problem we proceed in two steps: first we construct a feasible solution for the program (D) which yields a lower bound for the TNMU, and in the second step we construct a sequence of segments that realizes this lower bound. We define a directed acyclic graph $\overrightarrow{G} = (V \cup \{0,1\}, E)$. For E we take all possible arcs of the forms $(0,(i,1))$ and $((i,n),1)$, as well as all the arcs between the $j-$th and the $(j+1)-$th column $(j = 1,2,\ldots,n-1)$, precisely $E = E_1 \cup E_2 \cup E_3$, where

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

The next step is the definition of a weight function δ (depending on A) on E. δ reflects the structure of certain dual feasible solutions g (to be defined in the next section), in such a way that the objective value of the program

(D) for a solution g equals the weight of a certain $(0,1)$–path in $\overrightarrow{G}$. We put $d_{i,j} = a_{i,j} - a_{i,j-1}$ $(i \in [m],\ j \in [n+1])$ and

$$\begin{aligned}
\delta(0,(i,1)) &= a_{i,1} && (i \in [m]),\\
\delta((i,n),1) &= 0 && (i \in [m]),\\
\delta((i,j),(i,j+1)) &= \max\{0, d_{i,j+1}\} && (i \in [m], j \in [n-1]),\\
\delta((i,j),(i',j+1)) &= \max\{0, d_{i',j+1}\} - \sum_{k=i}^{i'-1} a_{k,j} \\
&\qquad\qquad (i,i' \in [m], i < i', j \in [n-1]),\\
\delta((i,j),(i',j+1)) &= \max\{0, d_{i',j+1}\} - \sum_{k=i'+1}^{i} a_{k,j} \\
&\qquad\qquad (i,i' \in [m], i > i', j \in [n-1]).
\end{aligned}$$

Since the considered matrix will be clear from the context we omit it in the notation for the weight function. For a path $P = (v_0, v_1, \dots, v_l)$ in $\overrightarrow{G}$ its weight is $\delta(P) = \sum_{i=1}^{l} \delta(v_{i-1}, v_i)$. Now we are prepared to formulate the main result of this chapter.

Theorem 1. *The minimal TNMU of a segmentation of a nonnegative matrix A equals the maximal weight of a $(0,1)-$path in $\overrightarrow{G}$.*

Note that the minimal TNMU in a segmentation without ICC can be interpreted analogously. In this case the minimal TNMU equals (see [10])

$$\max_{1 \le i \le m} \sum_{j=1}^{n} \max\{0, d_{i,j}\},$$

that is the maximal weight of a $(0,1)-$path in the graph that is obtained from $\overrightarrow{G}$ by deleting all the arcs $((i,j),(i',j+1))$ with $i \neq i'$. For notational convenience we put

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

so that in order to prove the theorem we have to show that $c(A)$ is a lower bound for the TNMU of a segmentation of A and that this bound is sharp.

2.2 The lower bound

In this section we show how the $(0,1)-$paths in $\overrightarrow{G}$ correspond to certain feasible solutions for the program (D) and from this we derive the lower bound part of Theorem 1. A $(0,1)-$path P is uniquely determined by the indices of the columns in which P changes the row and the indices of the rows in which P runs between the row changes. So let $x_1, x_2, \dots, x_{k-1}$ with $0 < x_1 < x_2 < \dots < x_{k-1} < n$ denote the indices of the columns where P changes the row, i.e.

$$(i, x_t), (i', x_t + 1) \in P \quad \text{with} \quad i \neq i' \quad (t \in [k-1]),$$

and let i_t^* be the row index with $(i_t^*, x_t) \in P$ $(t = 1, 2, \dots, k-1)$ and i_k^* the index with $(i_k^*, n) \in P$. Finally, we put $x_0 = 0$, and $x_k = n + 1$. Thus

$$P = (0, (i_1^*, 1), (i_1^*, 2), \dots, (i_1^*, x_1), (i_2^*, x_1 + 1), \dots, (i_2^*, x_2), \dots, \dots, (i_k^*, n), 1),$$

and P is uniquely determined by its parameters $(i_1^*, x_1), \dots, (i_k^*, x_k)$. Now we define $g : V \to \{1, -1, 0\}$ by

$$g(i,j) = \begin{cases} 1 & \text{if } x_{t-1} < j < x_t - 1,\ i = i_t^*,\ d_{i,j} \geq 0,\ d_{i,j+1} < 0, \\ 1 & \text{if } x_{t-1} < j = x_t - 1,\ i = i_t^*,\ d_{i,j} \geq 0, \\ -1 & \text{if } x_{t-1} < j < x_t - 1,\ i = i_t^*,\ d_{i,j} < 0,\ d_{i,j+1} \geq 0, \\ -1 & \text{if } j = x_t,\ i_t^* \leq i < i_{t+1}^* \text{ or } i_{t+1}^* < i \leq i_t^*, \\ -1 & \text{if } j = x_t,\ i = i_{t+1}^*,\ d_{i,j+1} \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

For $i \in [m]$ we put $J(i) := \{j \in [n] \ : \ g(i,j) \neq 0\}$. Fix some $i \in [m]$ and denote the elements of $J(i)$ by $j_1, j_2, \dots, j_p$ such that

$$j_1 < j_2 < \dots < j_p.$$

Then the following observations follow immediately from (2.5).

1. If $i = i_1^*$ and $k > 1$ then $x_1 \in J(i)$ and the sequence

   $$g(i, j_1), g(i, j_2), \dots, g(i, x_1)$$

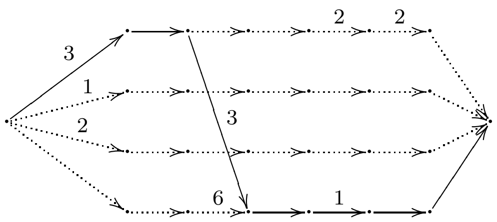
   is an alternating $(1,-1)-$sequence ending with -1.

2. If $i = i_k^*$, $k > 1$ and $J(i) \cap \{x_{k-1}, x_{k-1} + 1, \dots, n\} \neq \emptyset$ then for

   $$q = \min\{\tau \ : \ j_\tau \geq x_{k-1}\},$$

   $g(i, j_q), g(i, j_{q+1}), \dots, g(i, j_p)$ is an alternating $(1,-1)-$sequence starting with -1 and ending with 1.

3. If $i = i_1^*$ and $k = 1$ then the sequence $g(i, j_1), g(i, j_2), \dots, g(i, j_p)$ is empty or an alternating $(1,-1)-$sequence starting and ending with 1.



$$(g_{i,j}) = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 & 1 \end{pmatrix}$$

Fig. 2.1: A path P and the corresponding g.

4. If $i = i_t^*$ for $2 \leq t \leq k-1$ then $x_t \in J(i)$ and for

$$q = \min\{\tau \ : \ j_\tau \geq x_{t-1}\},$$

$g(i, j_q), g(i, j_{q+1}), \ldots, g(i, x_t)$ is an alternating $(1, -1)-$sequence starting and ending with -1.

5. If $j \in J(i)$ and (i, j) does not correspond to a term in one of the sequences described in the first 4 cases then $j = x_t$ for some $t \in [k-1]$ with $i \neq i_t^*$ and $i \neq i_{t+1}^*$ and $g(i, j) = -1$.

Example 2. Fig. 2.1 shows a path P of weight 7 with respect to the matrix

$$A = \begin{pmatrix} 3 & 0 & 0 & 0 & 2 & 4 \\ 1 & 1 & 1 & 2 & 3 & 3 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ 0 & 0 & 6 & 0 & 1 & 1 \end{pmatrix}$$

and the corresponding function g. The dotted lines are some more arcs labeled with their weight, where the unlabeled arcs have weight 0.

In order to prove that for every $(0, 1)-$path P the corresponding function g is a feasible solution for the program (D), we have to show that, for every $T \in \mathcal{F}$,

$$\sum_{(i,j) \in T} g(i, j) \leq 1.$$

Lemma 1. *Let P be a $(0, 1)-$path with parameters $(i_1^*, x_1), \ldots, (i_k^*, x_k)$, and let g be defined according to* (2.5)*. In addition let $1 \leq l \leq r+1 \leq n+1$. Then, for every $i \in [m]$,*

$$\sum_{j=l}^{r} g(i, j) \leq 1,$$

and equality implies $x_{t-1} < l \leq r < x_t$ for some $t \in [k]$ with $i_t^* = i$.

Proof. We choose an arbitrary $i \in [m]$. As above we denote the elements of $J(i) = \{j \in [n] \ : \ g(i,j) \neq 0\}$ by $j_1, \dots, j_p$ such that

$$j_1 < j_2 < \dots < j_p.$$

As a consequence of the observations before Example 2 we obtain, for $1 \leq q \leq p-1$,

$$g(i, j_q) = 1 \ \Rightarrow \ g(i, j_{q+1}) = -1.$$

Now the first part of the lemma follows from

$$\sum_{j=l}^{r} g(i,j) = \sum_{\tau=q}^{q'} g(i, j_\tau) \quad \text{for some } q, q' \in [p].$$

Suppose

$$\sum_{j=l}^{r} g(i,j) = \sum_{\tau=q}^{q'} g(i, j_\tau) = 1.$$

By construction the sequence $g(i, j_q), g(i, j_{q+1}), \dots, g(i, j_{q'})$ has to be an alternating $(1, -1)-$sequence starting and ending with 1. This implies

$$x_{t-1} < l < x_t \quad \text{and} \quad x_{t'-1} < r < x_{t'}$$

for some $t, t' \in [k]$ with $i = i_t^* = i_{t'}^*$. Assume $t \neq t'$ and put

$$t'' = \min\{\sigma > t \ : \ i_\sigma^* = i\} \quad \text{and} \quad q'' = \min\{\tau \ : \ j_\tau \geq x_{t''-1}\}.$$

Then

$$j_q < x_t < j_{q''} < j_{q'}, \quad g(i, x_t) = g(i, j_{q''}) = -1,$$

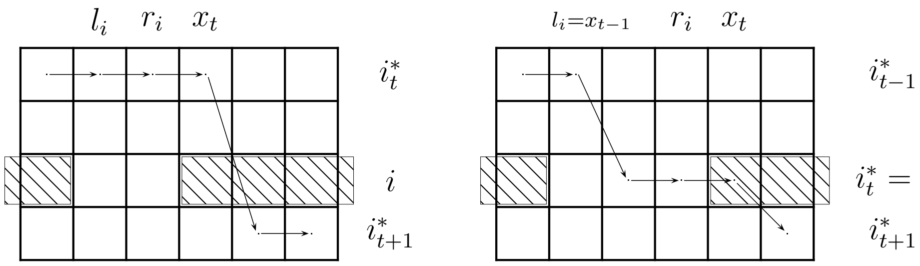
and $g(i,j) \leq 0$ for all j with $x_t < j < j_{q''}$. So $g(i, j_q), g(i, j_{q+1}), \dots, g(i, j_{q'})$ contains two consecutive $(-1)-$terms, which is a contradiction. Hence $t = t'$ and the second part of the lemma follows. ■

The next lemma gives a condition that must hold if the sum of the $g(i,j)$ over a row of a segment vanishes. (By a row of a segment we mean the part of the row that is left open by the MLC in the corresponding leaf position.)

Lemma 2. *Let P be a $(0,1)-$path with parameters $(i_1^*, x_1), \dots, (i_k^*, x_k)$, and let g be defined according to* (2.5)*. Assume $i \in [m]$, $1 \leq l \leq r+1 \leq n+1$ and*

$$\sum_{j=l}^{r} g(i,j) = 0.$$

Suppose in addition that for some $t \in [k-1]$ one of the following conditions holds



a) $i_t^* < i < i_{t+1}^*$ and $l \le x_t$ b) $t \ge 2$, $i_t^* = i < i_{t+1}^*$ and $l \le x_{t-1}$.

Fig. 2.2: Illustration of Lemma 2.

1. *$i_t^* < i < i_{t+1}^*$ and $l \le x_t$*
2. *$t \ge 2$, $i_t^* = i < i_{t+1}^*$ and $l \le x_{t-1}$*
3. *$i_t^* > i > i_{t+1}^*$ and $l \le x_t$*
4. *$t \ge 2$, $i_t^* = i > i_{t+1}^*$ and $l \le x_{t-1}$*

Then $r < x_t$.

Proof. We consider only the first two cases that are illustrated in Fig. 2.2. The other two are treated analogously. Assume $r \ge x_t$. In order to derive a contradiction we use the following observation several times. If P leaves row i in (i, j) then $g(i, j) = -1$, and if P enters row i' in (i', j'), $j' > 1$, then either $g(i', j' - 1) = -1$ or the first nonvanishing $g(i', j'')$ we meet on the subpath starting with (i', j') equals -1. We put

$$J = \{j \ : \ l \le j \le r,\ g(i, j) \ne 0\},$$

and denote the elements of J by $j_1, j_2, \dots, j_p$ ($j_1 < j_2 < \cdots < j_p$). In particular $j_q = x_t$ for some $q \in [p]$.

Case 1: $g(i, j_1) = -1$.

By assumption $g(i, j_1), \dots, g(i, j_p)$ is an alternating $(1, -1)-$sequence starting with -1 and ending with 1. From $g(i, x_t) = -1$ follows $q < p$ and by construction of g the contradiction

$$g(i, j_q) = g(i, j_{q+1}) = -1.$$

Case 2: $g(i, j_1) = 1$.

This implies $l < x_{t'}$ for some $t' < t$ with $i^*_{t'} = i$, and consequently $j_{q'} = x_{t'}$ for some $q' \in [p-1]$, $q' < q$. Thus

$$g(i, j_{q'}) = g(i, j_{q'+1}) = -1,$$

and $g(i,j_1), g(i,j_2), \dots, g(i,j_p)$ contains two consecutive $(-1)-$terms. By assumption this implies $g(i,j_p) = 1$, hence $p > q$, and by construction of g,

$$g(i, j_q) = g(i, j_{q+1}) = -1.$$

But now $g(i,j_1), g(i,j_2), \dots, g(i,j_p)$ contains two pairs of consecutive $(-1)-$terms (if $q'+1 < q$) or three consecutive $(-1)-$terms (if $q'+1 = q$). Again this yields a contradiction. ■

The following lemma is the crucial step in the proof of the feasibility of g. We show that the ICC implies that in any segment, between two rows in which the values of g add up to 1 there is a row in which this sum is at most -1.

Lemma 3. *Let P be a $(0,1)-$path with parameters $(i^*_1, x_1), \dots, (i^*_k, x_k)$, and let g be defined according to* (2.5). *Suppose S is a segment described by $l_1, l_2, \dots, l_m, r_1, r_2, \dots, r_m$ and there are row indices i_0, i_1 ($1 \le i_0 < i_1 \le m$) such that*

$$\sum_{j=l_{i_0}}^{r_{i_0}} g(i_0, j) = 1 \quad \textit{and} \quad \sum_{j=l_{i_1}}^{r_{i_1}} g(i_1, j) = 1.$$

Then there exists a row index i with $i_0 < i < i_1$ and $\sum_{j=l_i}^{r_i} g(i,j) \le -1$.

Proof. W.l.o.g. we may assume that there is no row i with $i_0 < i < i_1$ and

$$\sum_{j=l_i}^{r_i} g(i,j) = 1.$$

Suppose that for all i with $i_0 < i < i_1$, $\sum_{j=l_i}^{r_i} g(i,j) = 0$. By Lemma 1 there are $t, t' \in [k]$ such that

$$x_{t-1} < l_{i_0} \le r_{i_0} < x_t, \quad i^*_t = i_0 \quad \text{and}$$
$$x_{t'-1} < l_{i_1} \le r_{i_1} < x_{t'}, \quad i^*_{t'} = i_1.$$

W.l.o.g. we may assume $t < t'$. Now let $i_0 = z_0 < z_1 < \dots < z_p = i_1$ be an increasing sequence of row indices such that there is a corresponding

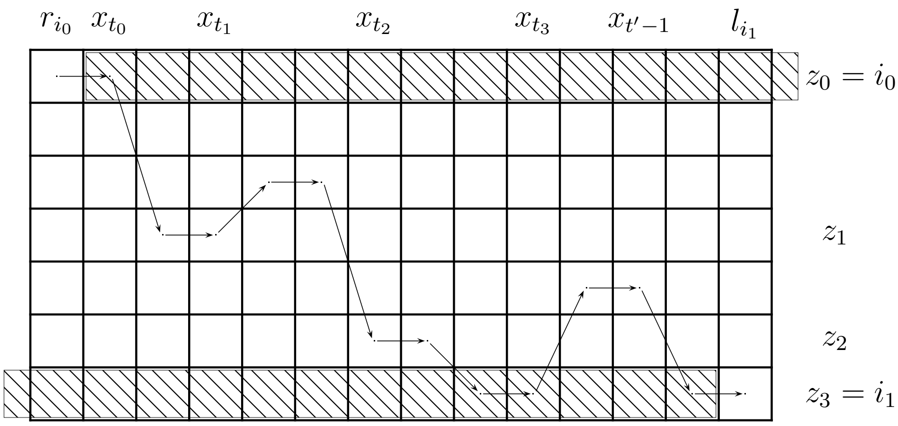


Fig. 2.3: Situation in the proof of Lemma 3 with $p = 3$.

sequence $t = t_0 < t_1 < \cdots < t_p \le t'$ with $i^*_{t_q} = z_q$ $(0 \le q \le p)$ and in addition for $0 \le q \le p-1$ there is no τ with $t_q < \tau < t_{q+1}$ and $z_q < i^*_\tau \le z_{q+1}$. These sequences always exist, are uniquely determined, and can be obtained recursively as follows. We put

$$t_0 = t \quad \text{and} \quad z_0 = i_0,$$

and for $q \ge 1$, if $z_{q-1} < i_1$,

$$t_q = \min\{\tau \;:\; z_{q-1} < i^*_\tau \le i_1\} \quad \text{and} \quad z_q = i^*_{t_q}.$$

So for some q we obtain $z_q = i_1$, and then we put $p = q$ (see Fig. 2.3).

Claim 1: For $0 \le q \le p-1$,

$$r_{z_q} < x_{t_q} \;\Rightarrow\; r_i < x_{t_{q+1}-1} \text{ for all } i \text{ with } z_q \le i < z_{q+1}.$$

Claim 2: For $0 \le q \le p-1$ we have $r_{z_q} < x_{t_q}$.

Proof of Claim 1. We proceed by induction on i. By assumption

$$r_{z_q} < x_{t_q} \le x_{t_{q+1}-1}.$$

So let $z_q < i < z_{q+1}$ and assume $r_{i-1} < x_{t_{q+1}-1}$. The ICC implies

$$l_i \le r_{i-1} + 1 \le x_{t_{q+1}-1},$$

and by Lemma 2 we obtain $r_i < x_{t_{q+1}-1}$. □

Proof of Claim 2. Here we use induction on q. Clearly,

$$r_{z_0} = r_{i_0} < x_t = x_{t_0}.$$

So let $q > 0$ and assume by induction $r_{z_{q-1}} < x_{t_{q-1}}$. Then by Claim 1,

$$r_{z_q-1} < x_{t_q-1}.$$

Thus by the ICC

$$l_{z_q} \leq r_{z_q-1} + 1 \leq x_{t_q-1},$$

and hence, again by Lemma 2, $r_{z_q} < x_{t_q}$. $\square$

Combining Claims 1 and 2 we obtain

$$r_{i_1-1} < x_{t_p-1} \leq x_{t'-1} < l_{i_1},$$

thus $r_{i_1-1} < l_{i_1} - 1$ in contradiction to the ICC. $\blacksquare$

Lemma 4. *Let P be a $(0,1)-$path with parameters $(i_1^*, x_1), \ldots, (i_k^*, x_k)$, and let g be defined according to* (2.5)*. Then g is feasible for* (D)*.*

Proof. Let $T \in \mathcal{F}$ be arbitrary and let S be the corresponding segment with parameters l_i, r_i $(i \in [m])$. Then

$$\sum_{(i,j)\in T} g(i,j) = \sum_{i=1}^{m} \sum_{j=l_i}^{r_i} g(i,j).$$

By Lemma 1, for all $i \in [m]$, $\sum_{j=l_i}^{r_i} g(i,j) \leq 1$, and by Lemma 3 between two rows i and i'' with $i < i''$ and

$$\sum_{j=l_i}^{r_i} g(i,j) = \sum_{j=l_{i''}}^{r_{i''}} g(i'',j) = 1$$

there is always a row i' with $i < i' < i''$ and $\sum_{j=l_{i'}}^{r_{i'}} g(i',j) \leq -1$. Consequently,

$$\sum_{(i,j)\in T} g(i,j) \leq 1,$$

that is the feasibility of g. $\blacksquare$

Lemma 5. *Let P be a $(0,1)-$path with parameters $(i_1^*, x_1), \dots, (i_k^*, x_k)$, and let g be defined according to* (2.5)*. Then*

$$\sum_{(i,j)\in V} g(i,j)a_{ij} = \sum_{t=1}^{k} \sum_{j=x_{t-1}+1}^{x_t-1} \max\{0, d_{i_t^*,j}\} - \sum_{t=1}^{k-1} \left(\sum_{i=i_t^*}^{i_{t+1}^*-1} a_{i,x_t} + \sum_{i=i_{t+1}^*+1}^{i_t^*} a_{i,x_t} \right).$$

For brevity of notation, here and for the rest of this work we use the convention that an empty sum is zero, i.e. $\sum_{i=r}^{s} z_i = 0$ if $s < r$.

Proof. Immediately from (2.5) it follows that

$$\sum_{j=x_{t-1}}^{x_t-1} g(i_t^*, j)a_{i_t^*,j} = \sum_{j=x_{t-1}+1}^{x_t-1} \max\{0, d_{i_t^*,j}\} \quad (t = 1, 2, \dots, k).$$

The remaining nonzero $g(i,j)$ correspond to the row changes of P, and we have to add for $t = 1, 2, \dots, k-1$,

$$\sum_{i=i_t^*}^{i_{t+1}^*-1} g(i, x_t)a_{i,x_t} = - \sum_{i=i_t^*}^{i_{t+1}^*-1} a_{i,x_t} \qquad \text{if } i_t^* < i_{t+1}^* \quad \text{and}$$

$$\sum_{i=i_{t+1}^*+1}^{i_t^*} g(i, x_t)a_{i,x_t} = - \sum_{i=i_{t+1}^*+1}^{i_t^*} a_{i,x_t} \qquad \text{if } i_t^* > i_{t+1}^*. \qquad \blacksquare$$

For the weight of P to be equal to the value of the program (D) for the corresponding g we need an additional restriction on P. We call the $(0,1)-$path P with parameters $(i_1^*, x_1), \dots, (i_k^*, x_k)$ feasible (with respect to A) if $d_{i_t^*,x_t} < 0$ for $t = 1, 2, \dots, k-1$, which in particular implies that the last arcs of the horizontal parts of P have weight 0.

Lemma 6. *Let P be a feasible $(0,1)-$path and let g be defined according to* (2.5)*. Then*

$$\sum_{(i,j)\in V} g(i,j)a_{i,j} = \delta(P).$$

Proof. Let P be given by parameters $(i_1^*, x_1), \dots, (i_k^*, x_k)$. For $t \in [k]$ we denote by P_t the subpath from $(i_t^*, x_{t-1}+1)$ to (i_t^*, x_t). Thus

$$\delta(P) = \sum_{t=1}^{k} \delta(P_t) + \delta(0, (i_1^*, 1)) + \sum_{t=1}^{k-1} \delta((i_t^*, x_t), (i_{t+1}^*, x_t + 1)).$$

From the feasibility of P follows that the last arc of P_t has weight 0 for all $t \in [k]$, and we obtain

$$\delta(P_t) = \sum_{j=x_{t-1}+2}^{x_t-1} \max\{0, d_{i_t^*,j}\}.$$

In addition, $\delta(0,(i_1^*,1)) = a_{i_1^*,1} = \max\{0, d_{i_1^*,1}\}$, and for $t \in [k-1]$,

$$\delta((i_t^*,x_t),(i_{t+1}^*,x_t+1)) = \max\{0, d_{i_{t+1}^*,x_t+1}\} - \sum_{i=i_t^*}^{i_{t+1}^*-1} a_{i,x_t} - \sum_{i=i_{t+1}^*+1}^{i_t^*} a_{i,x_t}.$$

Thus

$$\delta(P) = \sum_{t=1}^{k} \sum_{j=x_{t-1}+1}^{x_t-1} \max\{0, d_{i_t^*,j}\} - \sum_{t=1}^{k-1} \left(\sum_{i=i_t^*}^{i_{t+1}^*-1} a_{i,x_t} + \sum_{i=i_{t+1}^*+1}^{i_t^*} a_{i,x_t} \right),$$

and the claim follows by Lemma 5. ∎

Lemma 7. *There exists a feasible* $(0,1)-$*path* P *with* $\delta(P) = c(A)$.

Proof. For any $(0,1)-$path P with parameters $(i_1^*,x_1),\dots,(i_k^*,x_k)$ denote by $R(P) \subseteq [k-1]$ the subset of indices that destroy the feasibility of P, i.e.

$$R(P) = \{t \in [k-1] : \ d_{i_t^*,x_t} \geq 0\}.$$

Then

$$\lambda(P) = \sum_{t\in R(P)} \left| i_t^* - i_{t+1}^* \right|$$

measures how far P is from being feasible. In particular, $\lambda(P) = 0$ is a necessary and sufficient condition for the feasibility of P. Let P_0 be a $(0,1)-$path with parameters $(i_1^*,x_1),\dots,(i_k^*,x_k)$ and weight $\delta(P_0) = c(A)$. If $\lambda(P_0) = 0$ then P_0 is feasible and there is nothing to do. So we assume that for $r \geq 1$ we have a $(0,1)-$path P_{r-1} with parameters

$$(i_1^*,x_1),\dots,(i_k^*,x_k),$$

$\delta(P_{r-1}) = c(A)$ and $\lambda(P_{r-1}) > 0$. From this we construct a $(0,1)-$path P_r with $\delta(P_r) = c(A)$ and $\lambda(P_r) \leq \lambda(P_{r-1}) - 1$. This will prove the lemma, since after finitely many steps we obtain a path P with $\delta(P) = c(A)$ and $\lambda(P) = 0$. Let t be the smallest element of $R(P_{r-1})$.

Case 1: $d_{i_t^*,j} \geq 0$ for $x_{t-1} < j < x_t$.

We define P_r as follows.

1. If $i_t^* < i_{t+1}^* - 1$ and $i_{t-1}^* \neq i_t^* + 1$ the parameters of P_r are (see Fig. 2.4 and 2.6)

$$(i_1^*, x_1), \ldots, (i_{t-1}^*, x_{t-1}), (i_t^* + 1, x_t), (i_{t+1}^*, x_{t+1}), \ldots, (i_k^*, x_k).$$

2. If $i_t^* > i_{t+1}^* + 1$ and $i_{t-1}^* \neq i_t^* - 1$ the parameters of P_r are (see Fig. 2.9 and 2.11)

$$(i_1^*, x_1), \ldots, (i_{t-1}^*, x_{t-1}), (i_t^* - 1, x_t), (i_{t+1}^*, x_{t+1}), \ldots, (i_k^*, x_k).$$

3. If $i_{t-1}^* - 1 = i_t^* < i_{t+1}^* - 1$ or $i_{t-1}^* + 1 = i_t^* > i_{t+1}^* + 1$ the parameters of P_r are (see Fig. 2.7 and 2.12)

$$(i_1^*, x_1), \ldots, (i_{t-2}^*, x_{t-2}), (i_{t-1}^*, x_t), (i_{t+1}^*, x_{t+1}), \ldots, (i_k^*, x_k).$$

4. If $i_t^* + 1 = i_{t+1}^* \neq i_{t-1}^*$ or $i_t^* - 1 = i_{t+1}^* \neq i_{t-1}^*$ the parameters of P_r are (see Fig. 2.5 and 2.10)

$$(i_1^*, x_1), \ldots, (i_{t-1}^*, x_{t-1}), (i_{t+1}^*, x_{t+1}), \ldots, (i_k^*, x_k).$$

5. If $i_t^* + 1 = i_{t+1}^* = i_{t-1}^*$ or $i_t^* - 1 = i_{t+1}^* = i_{t-1}^*$ the parameters of P_r are (see Fig. 2.8 and 2.13)

$$(i_1^*, x_1), \ldots, (i_{t-2}^*, x_{t-2}), (i_{t+1}^*, x_{t+1}), \ldots, (i_k^*, x_k).$$

Case 2: $d_{i_t^*,j} < 0$ for some j with $x_{t-1} < j < x_t$.

We put

$$x := \max\{j \leq x_t \ : \ d_{i_t^*,j} < 0, \ d_{i_t^*,j+1} \geq 0\},$$

and define P_r as follows.

1. If $i_t^* < i_{t+1}^* - 1$ the parameters of P_r are (see Fig. 2.14)

$$(i_1^*, x_1), \ldots, (i_{t-1}^*, x_{t-1}), (i_t^*, x), (i_t^* + 1, x_t), (i_{t+1}^*, x_{t+1}), \ldots, (i_k^*, x_k).$$

2. If $i_t^* > i_{t+1}^* + 1$ the parameters of P_r are (see Fig. 2.16)

$$(i_1^*, x_1), \ldots, (i_{t-1}^*, x_{t-1}), (i_t^*, x), (i_t^* - 1, x_t), (i_{t+1}^*, x_{t+1}), \ldots, (i_k^*, x_k).$$

3. If $i_t^* = i_{t+1}^* - 1$ or $i_t^* = i_{t+1}^* + 1$ the parameters of P_r are (see Fig. 2.15 and 2.17)

$$(i_1^*, x_1), \dots, (i_{t-1}^*, x_{t-1}), (i_t^*, x), (i_{t+1}^*, x_{t+1}), \dots, (i_k^*, x_k).$$

We have to show that $\delta(P_r) = c(A)$ and $\lambda(P_r) \le \lambda(P_{r-1}) - 1$. The last assertion follows from the fact that either

$$R(P_r) = R(P_{r-1}) \quad \text{or} \quad R(P_r) = R(P_{r-1}) \setminus \{t\},$$

and consequently,

$$\lambda(P_r) = \lambda(P_{r-1}) - 1 \quad \text{or} \quad \lambda(P_r) = \lambda(P_{r-1}) - \left| i_t^* - i_{t+1}^* \right|.$$

Now we check that in any case $\delta(P_r) \ge \delta(P_{r-1})$ and hence $\delta(P_r) = c(A)$. In the following let the vertices of $\overrightarrow{G}$ be denoted as in the corresponding figures. In addition for two vertices X and Y on a path P we denote by $D_P(X, Y)$ the weight of the $(X, Y)-$subpath of P. Then in any case,

$$\begin{aligned}\delta(P_r) = \delta(P_{r-1}) - D_{P_{r-1}}(U, A) - D_{P_{r-1}}(A, B) - D_{P_{r-1}}(B, V) \\ + D_{P_r}(U, A') + D_{P_r}(A', B') + D_{P_r}(B', V).\end{aligned} \tag{2.6}$$

Cases 1.1(a), 1.4(a): (Fig. 2.4, 2.5)

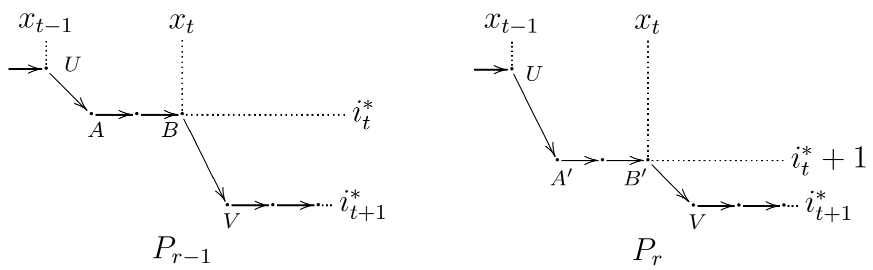


Fig. 2.4: Transition from P_{r-1} to P_r in Case 1.1.a) $i_{t-1}^* < i_t^*$.

Using $d_{i_t^*, j} \ge 0$ for $x_{t-1} < j \le x_t$ we obtain

$$\begin{aligned} D_{P_{r-1}}(A, B) &= a_{i_t^*, x_t} - a_{i_t^*, x_{t-1}+1}, \\ D_{P_r}(B', V) &= D_{P_{r-1}}(B, V) + a_{i_t^*, x_t}, \\ D_{P_r}(U, A') &= D_{P_{r-1}}(U, A) - \left(a_{i_t^*, x_{t-1}+1} - a_{i_t^*, x_{t-1}}\right) \\ &\qquad - a_{i_t^*, x_{t-1}} + \max\{0, d_{i_t^*+1, x_{t-1}+1}\} \\ &= D_{P_{r-1}}(U, A) - a_{i_t^*, x_{t-1}+1} + \max\{0, d_{i_t^*+1, x_{t-1}+1}\}. \end{aligned}$$

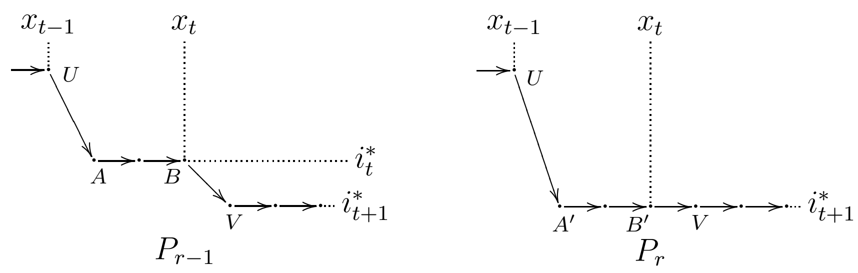


Fig. 2.5: Transition from P_{r-1} to P_r in Case 1.4.a) $i_{t-1}^* < i_t^*$.

Substituting into (2.6) yields

$$\begin{aligned}\delta(P_r) = \delta(P_{r-1}) - D_{P_{r-1}}(U,A) - \left(a_{i_t^*,x_t} - a_{i_t^*,x_{t-1}+1}\right) - D_{P_{r-1}}(B,V) \\ + \left(D_{P_{r-1}}(U,A) - a_{i_t^*,x_{t-1}+1} + \max\{0, d_{i_t^*+1,x_{t-1}+1}\}\right) \\ + D_{P_r}(A',B') + \left(D_{P_{r-1}}(B,V) + a_{i_t^*,x_t}\right),\end{aligned}$$

that is

$$\begin{aligned}\delta(P_r) &= \delta(P_{r-1}) + \max\{0, d_{i_t^*+1,x_{t-1}+1}\} + D_{P_r}(A',B') \\ &\geq \delta(P_{r-1}).\end{aligned}$$

Cases 1.1(b), 1.3(a), 1.5(a): (Fig. 2.6, 2.7, 2.8)

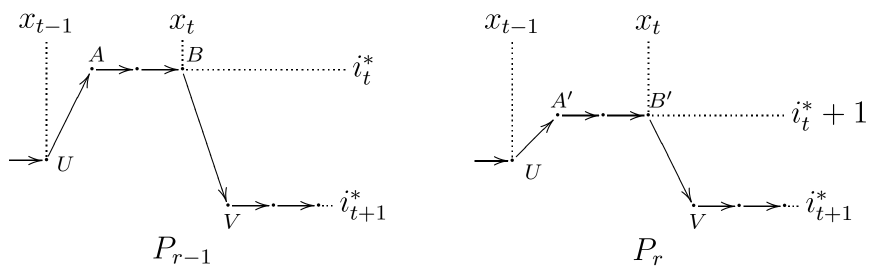


Fig. 2.6: Transition from P_{r-1} to P_r in Case 1.1.b) $i_{t-1}^* > i_t^*$.

Again,

$$\begin{aligned}D_{P_{r-1}}(A,B) &= a_{i_t^*,x_t} - a_{i_t^*,x_{t-1}+1}, \\ D_{P_r}(B',V) &= D_{P_{r-1}}(B,V) + a_{i_t^*,x_t}.\end{aligned}$$

But in these cases

$$\begin{aligned}D_{P_r}(U,A') = D_{P_{r-1}}(U,A) - \left(a_{i_t^*,x_{t-1}+1} - a_{i_t^*,x_{t-1}}\right) \\ + a_{i_t^*+1,x_{t-1}} + \max\{0, d_{i_t^*+1,x_{t-1}+1}\}.\end{aligned}$$

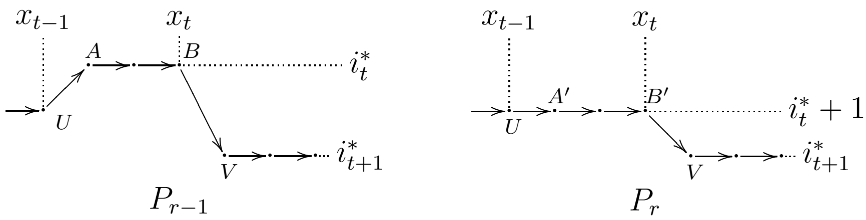


Fig. 2.7: Transition from P_{r-1} to P_r in Case 1.3.a) $i_{t-1}^* > i_t^*$.

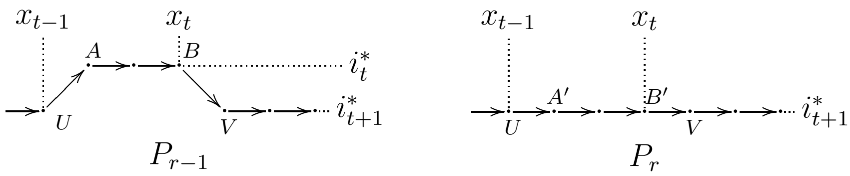


Fig. 2.8: Transition from P_{r-1} to P_r in Case 1.5.a) $i_{t-1}^* > i_t^*$.

And substituting into (2.6) yields

$$\begin{aligned}\delta(P_r) = \delta(P_{r-1}) - D_{P_{r-1}}(U,A) - \left(a_{i_t^*,x_t} - a_{i_t^*,x_{t-1}+1}\right) - D_{P_{r-1}}(B,V)\\ + \left[D_{P_{r-1}}(U,A) - \left(a_{i_t^*,x_{t-1}+1} - a_{i_t^*,x_{t-1}}\right) + a_{i_t^*+1,x_{t-1}} +\right.\\ \left.\max\{0, d_{i_t^*+1,x_{t-1}+1}\}\right] + D_{P_r}(A',B') + \left(D_{P_{r-1}}(B,V) + a_{i_t^*,x_t}\right),\end{aligned}$$

that is

$$\begin{aligned}\delta(P_r) &= \delta(P_{r-1}) + a_{i_t^*,x_{t-1}} + a_{i_t^*+1,x_{t-1}} + \max\{0, d_{i_t^*+1,x_{t-1}+1}\} + D_{P_r}(A',B')\\ &\geq \delta(P_{r-1}).\end{aligned}$$

Cases 1.2(a), 1.4(b): (Fig. 2.9, 2.10)

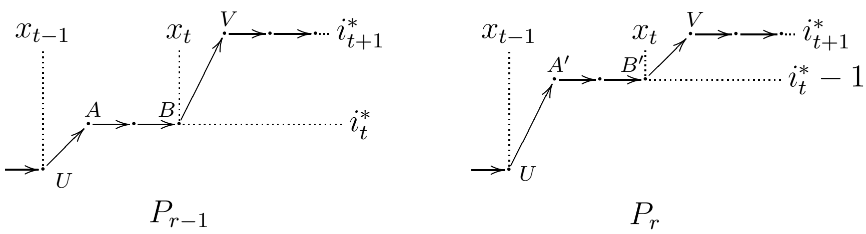


Fig. 2.9: Transition from P_{r-1} to P_r in Case 1.2.a) $i_{t-1}^* > i_t^*$.

The computation is the same as in Case 1.1(a) but in the formula for $D_{P_r}(U,A')$ we have to replace $d_{i_t^*+1,x_{t-1}+1}$ by $d_{i_t^*-1,x_{t-1}+1}$.

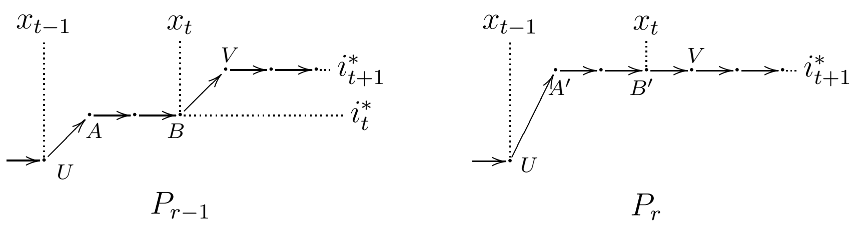


Fig. 2.10: Transition from P_{r-1} to P_r in Case 1.4.b) $i^*_{t-1} > i^*_t$.

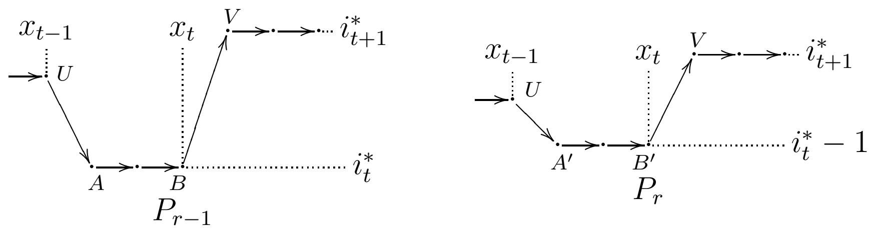


Fig. 2.11: Transition from P_{r-1} to P_r in Case 1.2.b) $i^*_{t-1} < i^*_t$.

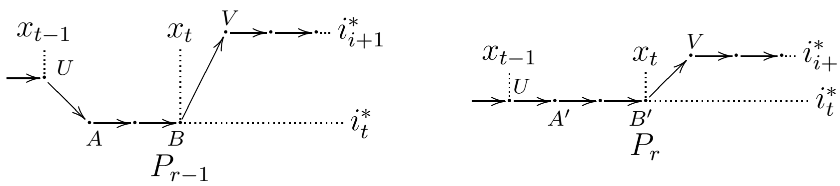


Fig. 2.12: Transition from P_{r-1} to P_r in Case 1.3.b) $i^*_{t-1} < i^*_t$.

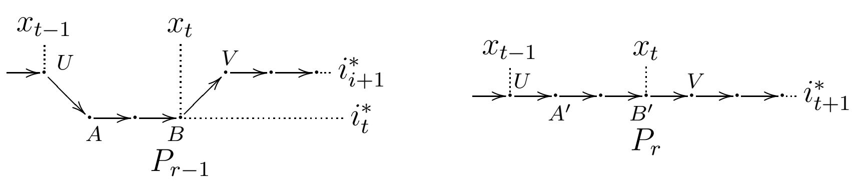


Fig. 2.13: Transition from P_{r-1} to P_r in Case 1.5.b) $i^*_{t-1} < i^*_t$.

Cases 1.2(b), 1.3(b), 1.5(b): (Fig. 2.11, 2.12, 2.13)

The computation is the same as in Case 1.1(b) but in the formula for $D_{P_r}(U, A')$ we have to replace $d_{i_t^*+1,x_{t-1}+1}$ by $d_{i_t^*-1,x_{t-1}+1}$.

Cases 2.1, 2.3(a): (Fig. 2.14, 2.15)

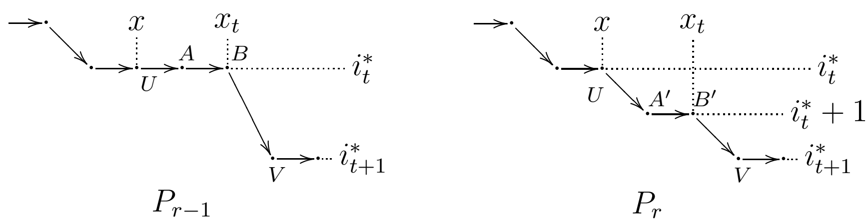


Fig. 2.14: Transition from P_{r-1} to P_r in Case 2.1.

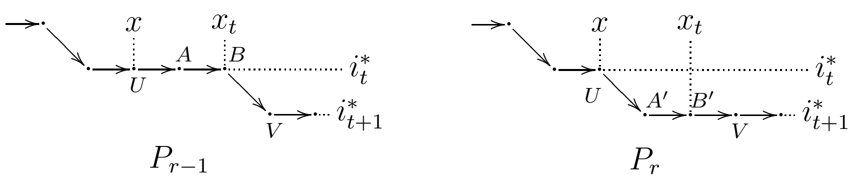


Fig. 2.15: Transition from P_{r-1} to P_r in Case 2.3.a) $i_{t-1}^* < i_t^*$.

Using $d_{i_t^*,j} \geq 0$ for $x < j < x_t$, we obtain

$$\begin{aligned}
D_{P_{r-1}}(U, A) &= a_{i_t^*,x+1} - a_{i_t^*,x}, \\
D_{P_{r-1}}(A, B) &= a_{i_t^*,x_t} - a_{i_t^*,x+1}, \\
D_{P_r}(B', V) &= D_{P_{r-1}}(B, V) + a_{i_t^*,x_t}, \\
D_{P_r}(U, A') &= \max\{0, d_{i_t^*+1,x+1}\} - a_{i_t^*,x} \\
&= \max\{0, d_{i_t^*+1,x+1}\} + D_{P_{r-1}}(U, A) - a_{i_t^*,x+1},
\end{aligned}$$

and so with (2.6)

$$\begin{aligned}
\delta(P_r) = \delta(P_{r-1}) - D_{P_{r-1}}(U, A) - \left(a_{i_t^*,x_t} - a_{i_t^*,x+1}\right) - D_{P_{r-1}}(B, V) \\
+ \left(\max\{0, d_{i_t^*+1,x+1}\} + D_{P_{r-1}}(U, A) - a_{i_t^*,x+1}\right) \\
+ D_{P_r}(A', B') + \left(D_{P_{r-1}}(B, V) + a_{i_t^*,x_t}\right),
\end{aligned}$$

that is

$$\begin{aligned}
\delta(P_r) &= \delta(P_{r-1}) + \max\{0, d_{i_t^*+1,x+1}\} + D_{P_r}(A', B') \\
&\geq \delta(P_{r-1}).
\end{aligned}$$

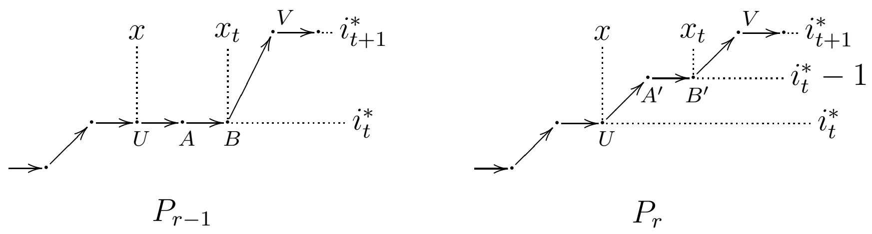


Fig. 2.16: Transition from P_{r-1} to P_r in Case 2.2.

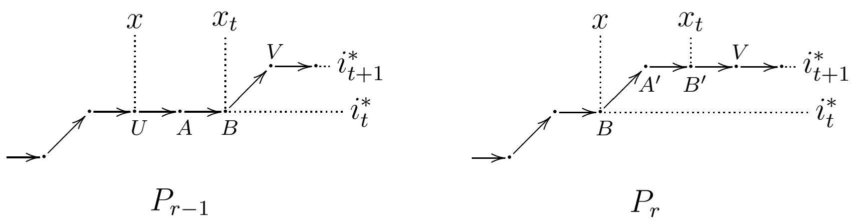


Fig. 2.17: Transition from P_{r-1} to P_r in Case 2.3.b) $i^*_{t-1} > i^*_t$.

Cases 2.2, 2.3(b): (Fig. 2.16, 2.17) The computation is the same as in Case 2.1 but in the formula for $D_{P_r}(U, A')$ we have to replace $d_{i^*_t+1,x+1}$ by $d_{i^*_t-1,x+1}$. ■

From Lemmas 4, 6 and 7 we deduce by duality that $c(A)$ is a lower bound for the sum of the coefficients of a segmentation of A and thus we have already proved the first half of the theorem.

2.3 The algorithm

In this section we assume $c(A) > 0$ and construct a segment S such that $A - S$ is still nonnegative and $c(A - S) \leq c(A) - 1$. Iterating this construction we obtain a sequence of $c(A)$ segments whose sum is A. For $(i, j) \in V$ we denote by $\alpha_1(i, j)$ the maximal weight of a $(0, (i, j))-$path, by $\alpha_2(i, j)$ the maximal weight of an $((i, j), 1)-$path and by $\alpha(i, j)$ the maximal weight of a $(0, 1)-$path through (i, j), that is

$$\begin{aligned}
\alpha_1(i,j) &= \max\{\delta(P) \ : \ P \ (0,(i,j)) - \text{path in } \overrightarrow{G}\},\\
\alpha_2(i,j) &= \max\{\delta(P) \ : \ P \ ((i,j),1) - \text{path in } \overrightarrow{G}\},\\
\alpha(i,j) &= \alpha_1(i,j) + \alpha_2(i,j).
\end{aligned}$$

Now we define two subsets $V_1, V_2 \subseteq V$. In V_1 we collect the pairs (i, j) that determine local maxima or right ends of plateaus in the sequences

$a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ $(i = 1, 2, \ldots, m)$, precisely

$$V_1 = \{(i,j) \in V \ : \ d_{i,j} \geq 0,\ d_{i,j+1} < 0\}.$$

The second subset V_2 is defined to be the set of pairs $(i,j) \in V_1$ with the following properties

1. There exists a $(0,1)-$path P of weight $c(A)$ through (i,j).
2. The sequence $a_{i,1}, \ldots, a_{i,j}$ is nondecreasing, i.e. $a_{i,1} \leq \cdots \leq a_{i,j}$.
3. The horizontal $(0,(i,j))-$path is a $(0,(i,j))-$path of maximal weight.

In other words,

$$V_2 = \{(i,j) \in V_1 \ : \ \alpha(i,j) = c(A) \text{ and } \alpha_1(i,j) = a_{i,j}\}.$$

Observe that for $(i,j) \in V_1$, $\delta((i,j),(i,j+1)) = 0$ and thus, for $j'' > j$,

$$\begin{aligned}
\delta((0,(i,1),(i,2),\ldots,(i,j''))) &= \sum_{j'=1}^{j''} \max\{0, d_{i,j'}\} \\
&\geq \sum_{j'=1}^{j} d_{i,j'} + \sum_{j'=j+2}^{j''} d_{i,j'} = a_{i,j} + (a_{i,j''} - a_{i,j+1}) \\
&> a_{i,j''},
\end{aligned}$$

and hence $\alpha_1(i,j'') > a_{i,j''}$. In particular, for any fixed row i there is at most one column index j with $(i,j) \in V_2$. In order to see that $c(A) > 0$ implies $V_2 \neq \emptyset$ consider a feasible $(0,1)-$path P with $\delta(P) = c(A)$. If P is a horizontal path without any row change then $\delta(P) > 0$ implies that P contains an element of V_1. Otherwise let $((i,j),(i',j+1))$ be the first row change of P. Then by the feasibility of P, $d_{i,j} < 0$ and thus the subpath $0,(i,1),\ldots,(i,j)$ contains an element of V_1. In both cases the first vertex on P which is in V_1 is in V_2 as well. Note that the $\alpha_1(i,j)$ $((i,j) \in [m] \times [n])$ can be determined as follows.

for $i = 1$ to m **do** $\quad \alpha_1(i,1) := a_{i,1}$
for $j = 2$ to n **do**
  for $i = 1$ to m **do** $\quad \alpha_1(i,j) := \max\limits_{1 \leq i' \leq m} \alpha_1(i',j-1) + \delta((i',j-1),(i,j))$

The $\alpha_2(i,j)$ can be determined analogously by running through the matrix from right to left. Obviously, this gives a method to determine $c(A)$ and the set V_2 in time $O(m^2 n)$. We denote the elements of V_2 by

$$(i_1,j_1),(i_2,j_2),\ldots,(i_t,j_t),$$

such that $i_1 < i_2 < \cdots < i_t$. A segment S (given by the parameters $l_1, l_2, \ldots, l_m, r_1, r_2, \ldots, r_m$) is constructed according to the following strategy. In row i_k ($k \in [t]$) we choose the open part maximal under the condition that the right boundary is j_k, i.e. we put

$$r_{i_k} = j_k \quad \text{and} \quad l_{i_k} = \max\{j \le j_k \ :\ a_{i_k,j} = 0\} + 1.$$

In the remaining rows we choose the open part minimal (in a sense to be made precise below) under the condition that the final result is a segment. The rows $i < i_1$ and $i > i_t$ remain closed. If $l_{i_k} > r_{i_{k+1}} + 1$ we choose the open part in row $i_k + 1$ maximal with $r_{i_k+1} = l_{i_k} - 1$. If necessary we repeat this step in the following rows, until finally $l_i \le r_{i_{k+1}} + 1$ for some i with $i_k < i < i_{k+1}$. If $l_{i_{k+1}} > r_{i_k} + 1$ we proceed analogously, starting in row $i_{k+1} - 1$. For the details of the construction see Algorithm 1.

Example 3. Let

$$A = \begin{pmatrix} 0&0&0&0&0&0&0&5&9\\ 0&0&0&0&1&1&2&4&2\\ 0&0&2&2&3&3&3&2&1\\ 1&1&2&2&1&1&1&1&1\\ 1&3&4&2&2&2&4&4&7\\ 2&2&2&2&1&2&2&3&3\\ 0&2&2&7&2&2&2&1&1 \end{pmatrix}.$$

Then $c(A) = 9$, $V_2 = \{(1,9),(5,3),(7,4)\}$ and the algorithm yields the segment

$$S = \begin{pmatrix} 0&0&0&0&0&0&0&1&\mathbf{1}\\ 0&0&0&0&1&1&1&0&0\\ 0&0&1&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 1&1&\mathbf{1}&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&1&1&\mathbf{1}&0&0&0&0&0 \end{pmatrix},$$

where the bold 1's correspond to the elements of V_2. For the resulting matrix

$$A - S = \begin{pmatrix} 0&0&0&0&0&0&0&4&8\\ 0&0&0&0&0&0&1&4&2\\ 0&0&1&1&3&3&3&2&1\\ 1&1&2&2&1&1&1&1&1\\ 0&2&3&2&2&2&4&4&7\\ 2&2&2&2&1&2&2&3&3\\ 0&1&1&6&2&2&2&1&1 \end{pmatrix}.$$

we have $c(A - S) = 8$.

Note that in Algorithm 1 each row i is considered exactly once and that l_i and r_i are defined either depending on some already defined values or by searching for a zero entry running through the row from right to left. So the total time complexity of Algorithm 1 is at most $O(mn)$. This is dominated by the construction of V_2, hence the complexity of the construction of S is $O(m^2 n)$. To prove the correctness of the algorithm we need an alternative description of paths in $\overrightarrow{G}$ that yields some insight into the relation between the constructed segment S and the path weights. For this let $\overrightarrow{H}$ be a directed

Algorithm 1 Segment

Input: $A = (a_{i,j})_{\substack{1 \le i \le m \\ 1 \le j \le n}}$ and $V_2 = \{(i_1, j_1), \ldots, (i_t, j_t)\}$
Output: l_i, r_i $(i = 1, \ldots, m)$
  for $(i, j) \in V_2$ **do**
    $l_i := \max\{j' \le j \ : \ a_{i,j'} = 0\}{+}1$
    $r_i := j$
  end for
  for $i = 1$ to $i_1 - 1$ **do**
    $l_i := l_{i_1}$; $r_i := l_i - 1$
  end for
  for $i = i_t + 1$ to m **do**
    $l_i := l_{i_t}$; $r_i := l_i - 1$
  end for
  for $k = 1$ to $t - 1$ **do**
    if $j_k > j_{k+1}$ **then**
      $i := i_k$
      while $i < i_{k+1}$ and $l_i > r_{i_{k+1}} + 1$ **do**
        $i := i + 1$
        $r_i := l_{i-1} - 1$
        $l_i := \max\{j \le r_i \ : \ a_{i,j} = 0\} + 1$
      end while
      for $i' = i + 1$ to $i_{k+1} - 1$ **do**
        $r_{i'} := r_{i_{k+1}}$; $l_{i'} := r_{i'} + 1$
      end for
    else
      $i := i_{k+1}$
      while $i > i_k$ and $l_i > r_{i_k} + 1$ **do**
        $i := i - 1$
        $r_i := l_{i+1} - 1$
        $l_i := \max\{j \le r_i \ : \ a_{i,j} = 0\} + 1$
      end while
      for $i' = i_k + 1$ to $i - 1$ **do**
        $r_{i'} := r_{i_k}$; $l_{i'} := r_{i'} + 1$
      end for
    end if
  end for
  return l_i, r_i $(i = 1, \ldots, m)$

graph with vertex set $V \cup \{0, 1\}$. As the arc set of H we take $E_0 = E_0^{(1)} \cup E_0^{(2)} \cup E_0^{(3)} \cup E_0^{(4)}$, where

$$\begin{aligned}
E_0^{(1)} &= \{(0, (i, 1)) \ : \ i \in [m]\} \cup \{((i, n), 1) \ : \ i \in [m]\}, \\
E_0^{(2)} &= \{((i, j), (i, j + 1) \ : \ i \in [m], \ j \in [n - 1])\}, \\
E_0^{(3)} &= \{((i, j), (i + 1, j)) \ : \ i \in [m - 1], \ j \in [n - 1]\}, \\
E_0^{(4)} &= \{((i, j), (i - 1, j)) \ : \ 2 \le i \le m, \ j \in [n - 1]\}.
\end{aligned}$$

Let the weight function δ_0 on E_0 be defined by

$$\begin{aligned}
\delta_0(0, (i, 1)) &= a_{i,1} && (i \in [m]), \\
\delta_0((i, n), 1) &= 0 && (i \in [m]), \\
\delta_0((i, j), (i, j + 1)) &= \max\{0, d_{i,j+1}\} && (i \in [m], \ j \in [n - 1]), \\
\delta_0((i, j), (i + 1, j)) &= -a_{i,j} && (i \in [m - 1], \ j \in [n - 1]), \\
\delta_0((i, j), (i - 1, j)) &= -a_{i,j} && (2 \le i \le m, \ j \in [n - 1]).
\end{aligned}$$

Example 4. Figure 2.18 shows $\overrightarrow{H}$ corresponding to the matrix

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

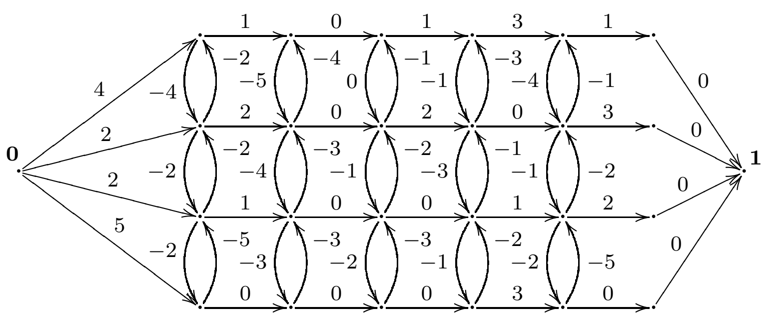


Fig. 2.18: The digraph corresponding to the matrix A in Example 4.

It is easy to see that there is a bijection between the paths in $\overrightarrow{G}$ and the paths in $\overrightarrow{H}$ with the additional restriction that the last arc is in $E_0^{(1)} \cup E_0^{(2)}$. In addition this bijection preserves the weight, that is for a path P in $\overrightarrow{G}$ and the corresponding path Q in $\overrightarrow{H}$ we have

$$\delta(P) = \delta_0(Q).$$

In particular, there is a weight–preserving bijection between the $(0,1)-$paths in $\overrightarrow{G}$ and $\overrightarrow{H}$. The advantage of Q compared to P is that possibly existing "long, skew" arcs in P are replaced by a sequence of vertical arcs and one horizontal arc, and the weights of these arcs are easier to control. Analogous to α, α_1 and α_2 we define for $(i,j) \in V$,

$$\begin{aligned}\beta_1(i,j) &= \max\{\delta_0(Q) \ : \ Q \ (0,(i,j)) - \text{path in } \overrightarrow{H}\},\\ \beta_2(i,j) &= \max\{\delta_0(Q) \ : \ Q \ ((i,j),1) - \text{path in } \overrightarrow{H}\},\\ \beta(i,j) &= \beta_1(i,j) + \beta_2(i,j).\end{aligned}$$

We need some information about the connection between the weights in $\overrightarrow{G}$ and $\overrightarrow{H}$. Obviously $\beta_2(i,j) = \alpha_2(i,j)$ for all $(i,j) \in V$. The next lemma is an analogous result about α_1 and β_1 for the vertices on $(0,1)-$paths of maximal weight.

Lemma 8. *For all $(i,j) \in V$ with $\alpha(i,j) = c(A)$, we have $\beta_1(i,j) = \alpha_1(i,j)$.*

Proof. $\beta_1(i,j) \geq \alpha_1(i,j)$ is trivial, since for every $(0,(i,j))-$path in $\overrightarrow{G}$ there is the corresponding $(0,(i,j))-$path in $\overrightarrow{H}$ of the same weight. Let Q_1 and Q_2 be a $(0,(i,j))-$path in $\overrightarrow{H}$ and an $((i,j),1)-$path in $\overrightarrow{H}$, respectively, with

$$\delta_0(Q_1) = \beta_1(i,j) \quad \text{and} \quad \delta_0(Q_2) = \beta_2(i,j) = \alpha_2(i,j).$$

By concatenating Q_1 and Q_2 we obtain a $(0,1)-$path Q in $\overrightarrow{H}$ with

$$\delta_0(Q) = \beta_1(i,j) + \alpha_2(i,j).$$

Since the last arc of Q is in $E_0^{(1)}$, this implies the existence of a $(0,1)-$path P in $\overrightarrow{G}$ with $\delta(P) = \beta_1(i,j) + \alpha_2(i,j)$. So

$$\beta_1(i,j) + \alpha_2(i,j) \leq c(A) = \alpha_1(i,j) + \alpha_2(i,j),$$

and thus $\beta_1(i,j) \leq \alpha_1(i,j)$. ■

Lemma 9. *Let $(i,j),(k,l) \in V$, $i > k$ and put $p = i - k$.*

1. *If $j < l$ and there are column indices $j'_1, j'_2, \ldots, j'_p$ such that*

$$j \leq j'_1 \leq j'_2 \leq \cdots \leq j'_p < l \quad \textit{and}$$
$$a_{i-q,j'_q} = 0 \qquad \textit{for } q = 1,2,\ldots,p,$$

then there exists a $((i,j),(k,l))-$path P in $\overrightarrow{G}$ with

$$\delta(P) \geq a_{k,l} - a_{i,j}.$$

$$\begin{array}{cccccccccc}
 & & j'_1 = j'_2 & j'_3 & & j'_4 & & \\
 & \vdots & & & & & \vdots & \\
\cdots & * & * & * & * & 0 & a_{k,l} & \cdots \\
 & * & * & 0 & * & * & * & \\
 & * & 0 & * & * & * & * & \\
 & * & 0 & * & * & * & * & \\
\cdots & a_{i,j} & * & * & * & * & * & \cdots \\
 & \vdots & & & & & \vdots &
\end{array}$$

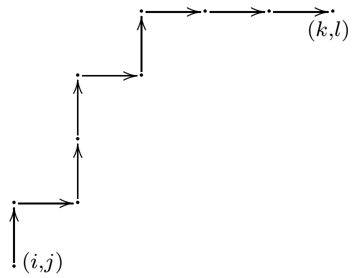


Fig. 2.19: A part of a matrix A and the corresponding path Q.

2. *If $j > l$ and there are column indices $j'_1, j'_2, \ldots, j'_p$ such that*

$$l \leq j'_1 \leq j'_2 \leq \cdots \leq j'_p < j \quad \textit{and}$$

$$a_{k+q,j'_q} = 0 \qquad \textit{for } q = 1, 2, \ldots, p,$$

then there exists a $((k,l),(i,j))-$path P in $\overrightarrow{G}$ with

$$\delta(P) \geq a_{i,j} - a_{k,l}.$$

Proof. We consider only the first case that is illustrated in Fig. 2.19. The second one is treated analogously. First we construct a $((i,j),(k,l))-$path Q in $\overrightarrow{H}$. We take $((i,j),(i-1,j))$ with weight $-a_{i,j}$ as the first arc and complete this arc to a $((i,j),(k,l))-$path Q in such a way, that row changes occur only along the arcs

$$((i-q, j'_q), (i-q-1, j'_q)) \qquad (1 \leq q \leq p-1).$$

This is possible by our assumption on the j'_q. Thus the vertical arcs of Q, except for the first one, have weight 0 and since the horizontal arcs have nonnegative weight in any case we conclude that the $((i,j),(k,j'_p))-$subpath of Q has weight at least $-a_{i,j}$. Finally the weight of the path

$$(k, j'_p), (k, j'_p + 1), \ldots, (k, l)$$

is at least $a_{k,l}$ and from $l > j'_p$ follows that the last arc of Q is in $E_0^{(2)}$ and thus there exists a $((i,j),(k,l))-$path P in $\overrightarrow{G}$ with

$$\delta(P) = \delta(Q) \geq a_{k,l} - a_{i,j}.$$

■

Lemma 10. *Algorithm 1 yields a segment S.*

Proof. Suppose the algorithm does not yield a segment. This is possible only if for some $k \in [t-1]$ the condition of the while–loop in line 14 (resp. line 24) holds for all $i \in \{i_k, i_k+1, \dots, i_{k+1}-1\}$ (resp. for all $i \in \{i_k+1, i_k+2, \dots, i_{k+1}\}$). If $j_k = j_{k+1}$ then

$$l_{i_k} \le r_{i_{k+1}} \quad \text{and} \quad l_{i_{k+1}} \le r_{i_k}.$$

So we may assume $j_k \ne j_{k+1}$. Let $j_k > j_{k+1}$. (The case $j_k < j_{k+1}$ can be treated analogously.) We put $p = i_{k+1} - i_k$ and

$$j'_q = l_{i_{k+1}-q} - 1 \qquad (q = 1, 2, \dots, p).$$

The assumption that the while–condition is fulfilled for all $i_{k+1} - q$ ($q = 1, 2, \dots, p$) implies

$$r_{i_{k+1}} + 1 \le j'_1 \le j'_2 \le \dots \le j'_p < j_k \quad \text{and}$$

$$a_{i_{k+1}-q, j'_q} = 0 \qquad (q = 1, 2, \dots, p).$$

Thus by Lemma 9 there is a $((i_{k+1}, j_{k+1}+1), (i_k, j_k))$–path P_0 in $\overrightarrow{G}$ of weight at least $a_{i_k,j_k} - a_{i_{k+1},j_{k+1}+1}$. Using $(i_{k+1}, j_{k+1}) \in V_2$, i.e. $a_{i_{k+1},j_{k+1}} > a_{i_{k+1},j_{k+1}+1}$, this yields

$$\delta_0(P_0) > a_{i_k,j_k} - a_{i_{k+1},j_{k+1}}.$$

Now we concatenate the path $0, (i_{k+1}, 1), (i_{k+1}, 2), \dots, (i_{k+1}, j_{k+1}+1)$ with P_0 to obtain a $(0, (i_k, j_k))$–path of weight at least

$$a_{i_{k+1},j_{k+1}} + \delta(P_0) > a_{i_k,j_k},$$

in contradiction to $(i_k, j_k) \in V_2$. ■

Let $S = (s_{i,j})$ be the result of Algorithm 1. By construction, $s_{i,j} = 1$ implies $a_{i,j} \ge 1$ and so the entries of $A - S$ are nonnegative. We put

$$\begin{aligned} a'_{i,j} &= a_{i,j} - s_{i,j} && (i \in [m], j \in [n]),\\ a'_{i,0} &= a_{i,n+1} = 0 && (i \in [m]),\\ d'_{i,j} &= a'_{i,j} - a'_{i,j-1} && (i \in [m], j \in [n]). \end{aligned}$$

By δ' and δ'_0 we denote the weight functions on $\overrightarrow{G}$ and $\overrightarrow{H}$, respectively, which correspond to $A' = (a'_{i,j})$. For $(i,j) \in V$ we put

$$\begin{aligned}
\alpha'_1(i,j) &= \max\{\delta'(P) \ : \ P\ (0,(i,j)) - \text{path in } \overrightarrow{G}\},\\
\alpha'_2(i,j) &= \max\{\delta'(P) \ : \ P\ ((i,j),1) - \text{path in } \overrightarrow{G}\},\\
\alpha'(i,j) &= \alpha'_1(i,j) + \alpha'_2(i,j),\\
\beta'_1(i,j) &= \max\{\delta'(Q) \ : \ Q\ (0,(i,j)) - \text{path in } \overrightarrow{H}\},\\
\beta'_2(i,j) &= \max\{\delta'(Q) \ : \ Q\ ((i,j),1) - \text{path in } \overrightarrow{H}\},\\
\beta'(i,j) &= \beta'_1(i,j) + \beta'_2(i,j).
\end{aligned}$$

By T we denote the subset of V which corresponds to the segment S, that is

$$T = \{(i,j) \in V \ : \ s_{i,j} = 1\}.$$

The next lemma asserts that for $(i,j) \in T$ the sequence $a_{i,1}, \dots, a_{i,j}$ is non-decreasing and the horizontal path from 0 to (i,j) has maximal weight with respect to A in both of $\overrightarrow{G}$ and $\overrightarrow{H}$.

Lemma 11. *For $(i,j) \in T$ we have*

$$\beta_1(i,j) = \alpha_1(i,j) = a_{i,j} \quad \textit{and} \quad \alpha(i,j) = c(A).$$

Proof. Let $(i,j) \in T$. Clearly, $\beta_1(i,j) \geq \alpha_1(i,j) \geq a_{i,j}$. Assume P_0 is a $(0,(i,j))$−path in $\overrightarrow{G}$ with $\delta(P_0) > a_{i,j}$. Recall that $V_2 = \{(i_1,j_1),\dots,(i_t,j_t)\}$ with $i_1 < i_2 < \cdots < i_t$. We claim that for some $k \in [t]$ there is an $((i,j),(i_k,j_k))$−path P_1 in $\overrightarrow{G}$ of weight at least $a_{i_k,j_k} - a_{i,j}$. To see this we distinguish three types of vertices in T:

1. $i = i_k$ and $j \leq j_k$ for some $k \in [t]$:

   The path $(i_k,j),(i_k,j+1),\dots,(i_k,j_k)$ has weight $a_{i_k,j_k} - a_{i_k,j}$.

2. $i_k < i < i_{k+1}$ for some $k \in [t-1]$ with $j_k > j_{k+1}$:

   By construction of S there are column indices $j'_1, j'_2, \dots, j'_p$, where $p = i - i_k$, such that

   $$j \leq j'_1 \leq j'_2 \leq \cdots \leq j'_p < j_k \quad \text{and}$$

   $$a_{i-q,j'_q} = 0 \qquad (q = 1,2,\dots,p).$$

   Thus the claim follows by Lemma 9.

3. $i_{k-1} < i < i_k$ for some $k \in \{2, 3, \ldots, t\}$ with $j_{k-1} < j_k$:

   By construction of S there are column indices $j'_1, j'_2, \ldots, j'_p$, where $p = i_k - i$, such that

$$j \leq j'_1 \leq j'_2 \leq \cdots \leq j'_p < j_k \quad \text{and}$$

$$a_{i+q, j'_q} = 0 \qquad (q = 1, 2, \ldots, p).$$

   Thus the claim follows by Lemma 9.

But now we can concatenate P_0 and P_1 to obtain a $(0, (i_k, j_k))$−path P in $\overrightarrow{G}$ with

$$\delta(P) = \delta(P_0) + \delta(P_1) > a_{i,j} + (a_{i_k,j_k} - a_{i,j}) = a_{i_k,j_k}$$

in contradiction to $(i_k, j_k) \in V_2$. This proves $\alpha_1(i, j) = a_{i,j}$. In addition, concatenating the paths $(0, (i, 1), (i, 2), \ldots, (i, j))$, P_1 and a $((i_k, j_k), 1)$−path of maximal weight yields $\alpha(i, j) = c(A)$ and thus also $\beta_1(i, j) = \alpha_1(i, j)$ by Lemma 8. ■

Now we want to prove that for $(i, j) \in T$ the horizontal $(0, (i, j))$−path is still maximal with respect to A'. We need the following necessary condition for $\beta_1(i, j) > a_{i,j}$.

Lemma 12. *Suppose $\beta_1(i, j) > a_{i,j}$ and Q is a $(0, (i, j))$−path in $\overrightarrow{H}$ with $\delta_0(Q) = \beta_1(i, j)$. Then there exists a vertex $(i', j') \in V_1$ such that either*

- *$j' = 1$ and $((i', 1), (i', 2))$ is an arc of Q or*
- *$1 < j' < n$ and $((i', j' - 1), (i', j'))$, $((i', j'), (i', j' + 1)$ are arcs of Q.*

If in addition $\beta(i, j) = c(A)$ then we can choose (i', j') even in V_2.

Proof. Let Q be a $(0, (i, j))$−path with $\delta_0(Q) = \beta_1(i, j)$ and assume there is no such vertex in V_1. We show $\delta_0(Q) = a_{i,j}$ which gives the desired contradiction. Clearly, $\delta_0(Q) \geq a_{i,j}$. The first arc of Q is of the form $(0, (i', 1))$ and has weight $a_{i',1}$. So we may assume that Q has more than one arc and proceed by induction on the number of arcs of an initial subpath of Q.

Case 1: The last arc of Q is in $E_0^{(3)} \cup E_0^{(4)}$.

W.l.o.g. the last arc is $((i - 1, j), (i, j))$ with weight $-a_{i-1,j}$. Since by induction $\delta_0(Q \setminus \{(i, j)\}) = a_{i-1,j}$, we obtain $\delta_0(Q) = 0 \leq a_{i,j}$.

Case 2: The last arc of Q is in $E_0^{(2)}$, and the second last arc is in $E_0^{(3)} \cup E_0^{(4)}$.
W.l.o.g. the last two arcs of Q are $((i - 1, j - 1), (i, j - 1))$ and $((i, j - 1), (i, j))$. By induction the weight of the $(0, (i - 1, j - 1))$−subpath of Q is $a_{i-1,j-1}$. Thus the weight of the $(0, (i, j - 1))$−subpath is 0 and by maximality of Q follows $a_{i,j-1} = 0$, hence $\delta_0(Q) = a_{i,j}$.

Case 3: The last two arcs of Q are in $E_0^{(1)} \cup E_0^{(2)}$.

By induction the $(0,(i,j-1))$−subpath of Q has weight $a_{i,j-1}$. By maximality of Q this implies $d_{i,j'} \geq 0$ for all j', $1 \leq j' \leq j-1$. Now $d_{i,j} \geq 0$, since otherwise $(i,j-1)$ is a vertex in V_1 that fulfills the conditions of the lemma. Thus $\delta_0(Q) = a_{i,j}$.

Now suppose $\beta(i,j) = c(A)$. Then we can complete Q to a $(0,1)$−path Q' of weight $c(A)$. Let P be the corresponding $(0,1)$−path in $\overrightarrow{G}$, and let $(i',j') \in V_1$ be the first vertex on Q that has the claimed properties. Then $(i',j') \in P$ and the $(0,(i',j'))$−subpath of P has weight $a_{i',j'}$, that is $(i',j') \in V_2$. ■

Lemma 13. *For $(i,j) \in T$ we have $\beta_1'(i,j) = \alpha_1'(i,j) = a_{ij}'$.*

Proof. Again trivially,

$$\beta_1'(i,j) \geq \alpha_1'(i,j) \geq a_{i,j}'.$$

Let $(i,j) \in T$ and assume $\beta_1'(i,j) > a_{i,j}'$. In particular, $j > 1$ since obviously $\beta_1'(i,1) = a_{i,1}'$ for all $i \in [m]$. There is a $(0,(i,j))$−path Q in $\overrightarrow{H}$ with

$$\delta_0'(Q) = \beta_1'(i,j) > a_{i,j}'.$$

W.l.o.g. we may assume that (i,j) is the first counterexample to the lemma on Q, i.e.

$$\beta_1'(i_0,j_0) = a_{i_0,j_0}' \quad \text{for all } (i_0,j_0) \in (Q \setminus \{(i,j)\}) \cap T.$$

Case 1: $(Q \setminus \{(i,j)\}) \cap T = \emptyset$.

Let e be the last arc of Q. Then $\delta_0(e_1) = \delta_0'(e_1)$ for all arcs $e_1 \neq e$ of Q.

Case 1.1: $e \in E_0^{(2)}$.

Then $\delta_0(e) = \delta_0'(e)+1$, hence $\delta_0(Q) = \delta_0'(Q)+1$, and consequently (using Lemma 11),

$$\beta_1'(i,j) = \delta_0(Q) - 1 \leq \beta_1(i,j) - 1 = a_{i,j}'.$$

Case 1.2: $e \in E_0^{(3)} \cup E_0^{(4)}$.

W.l.o.g. $e = ((i-1,j),(i,j))$ and $\delta_0(e) = \delta_0'(e) = -a_{i-1,j}$, and thus

$$\delta_0(Q) = \delta_0'(Q) = \beta_1'(i,j).$$

Assume $\delta_0(Q) = \beta_1(i,j) = a_{i,j}$. Then $\delta_0(Q) > 0$, and thus

$$\delta_0(Q \setminus \{(i,j)\}) > a_{i-1,j}.$$

By Lemma 11, $\beta(i,j) = \alpha(i,j) = c(A)$ and consequently by Lemma 12, $Q \setminus \{(i,j)\}$ contains a vertex $(i_0,j_0) \in V_2 \subseteq T$. This is a contradiction and we conclude

$$\beta_1'(i,j) = \delta_0(Q) < \beta_1(i,j) = a_{i,j},$$

and thus $\beta_1'(i,j) = a_{i,j}'$.

Case 2: $(Q \setminus \{(i,j)\}) \cap T \neq \emptyset$.

Let (i_0,j_0) be the last vertex on $Q \setminus \{(i,j)\}$ that is in T and denote by Q_1 and Q_2 the $(0,(i_0,j_0))-$subpath and the $((i_0,j_0),(i,j))-$subpath of Q, respectively. By assumption $\delta_0'(Q_1) = a_{i_0,j_0}'$, so w.l.o.g. we may assume $Q_1 = (0,(i_0,1),(i_0,2),\dots,(i_0,j_0))$, and then

$$\delta_0(Q_1) = \beta_1(i_0,j_0) = a_{i_0,j_0} = \delta_0'(Q_1) + 1. \tag{2.7}$$

We denote the arcs of Q_2 by $e_1,e_2,\dots,e_p$. For $p=1$ we obtain

$$\begin{aligned} \delta_0'(Q) &= \delta_0'(Q_1) - a_{i_0,j_0}' = 0 && \text{if } e_1 \in E_0^{(3)} \cup E_0^{(4)} \quad \text{and} \\ \delta_0'(Q) &= \delta_0'(Q_1) + \max\{0, d_{i,j}'\} && \text{if } e_1 \in E_0^{(2)}. \end{aligned}$$

Since $e_1 \in E_0^{(2)}$ implies $(i,j),(i,j-1) \in T$ and thus $d_{ij}' = d_{i,j} \geq 0$ (Lemma 11), we obtain $\delta_0'(Q) \leq a_{i,j}'$ and consequently $\beta_1'(i,j) = a_{i,j}'$. So let $p > 1$. Then

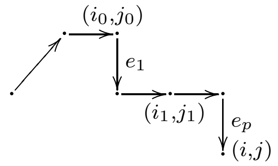
$$\delta_0'(e_i) = \delta_0(e_i) \quad (2 \leq i \leq p-1), \tag{2.8}$$

$$\delta_0'(e_1) = \begin{cases} \delta_0(e_1) + 1 & \text{if } e_1 \in E_0^{(3)} \cup E_0^{(4)}, \\ \delta_0(e_1) + 1 & \text{if } e_1 \in E_0^{(2)} \text{ and } d_{i_0,j_0+1} \geq 0, \\ \delta_0(e_1) & \text{if } e_1 \in E_0^{(2)} \text{ and } d_{i_0,j_0+1} < 0 \quad \text{and} \end{cases} \tag{2.9}$$

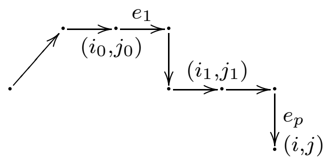
$$\delta_0'(e_p) = \begin{cases} \delta_0(e_p) & \text{if } e_p \in E_0^{(3)} \cup E_0^{(4)}, \\ \delta_0(e_p) - 1 & \text{if } e_p \in E_0^{(2)}, \end{cases} \tag{2.10}$$

and in particular,

$$\delta_0'(Q_2) \leq \delta_0(Q_2) + 1.$$



a) $e_1 \in E_0^{(3)} \cup E_0^{(4)}$



b) $e_1 \in E_0^{(2)}$ and $d_{i_0,j_0+1} \geq 0$

Fig. 2.20: Paths Q as in Case 2.2. of Lemma 13.

Case 2.1: $\delta_0'(Q_2) \leq \delta_0(Q_2)$.

$\delta_0'(Q) = \delta_0'(Q_1) + \delta_0'(Q_2) \leq a'_{i_0,j_0} + \delta_0(Q_2) < \delta_0(Q)$ implies

$$\beta_1'(i,j) < \beta_1(i,j) = a_{i,j},$$

and thus $\beta_1'(i,j) = a'_{i,j}$.

Case 2.2: $\delta_0'(Q_2) = \delta_0(Q_2) + 1$.

In this case (2.7) and (2.8)–(2.10) imply

$$\delta_0'(Q) = \delta_0(Q) \quad \text{and} \quad e_p \in E_0^{(3)} \cup E_0^{(4)},$$

w.l.o.g. $e_p = ((i-1,j),(i,j))$ with weight $-a_{i-1,j}$. Assume $\delta_0(Q) = \beta_1(i,j)$. Then $\delta_0(Q) > 0$ and thus

$$\beta_1(i-1,j) > a_{i-1,j}.$$

By Lemma 11, $\beta(i,j) = \alpha(i,j) = c(A)$, and by Lemma 12 there is a vertex $(i_1,j_1) \in V_2$ such that Q contains the arc $((i_1,j_1),(i_1,j_1+1))$. From (2.8)–(2.10) it follows that $\delta_0'(Q_2) = \delta_0(Q_2) + 1$ is possible only if

$$e_1 \in E_0^{(3)} \cup E_0^{(4)} \quad \text{or} \quad (e_1 \in E_0^{(2)} \text{ and } d_{i_0,j_0+1} \geq 0).$$

Hence, using $d_{i_0,j'} \geq 0$ for $1 \leq j' \leq j_0$, $(i_1,j_1) \notin Q_1$ and we obtain the contradiction

$$(i_1,j_1) \in (Q \setminus \{(i,j),(i_0,j_0)\}) \cap V_2.$$

Thus $\delta_0'(Q) = \delta_0(Q) < \beta_1(i,j) = a_{i,j}$, and so $\beta_1'(i,j) = a'_{i,j}$. ∎

Now we are prepared for the final step.

Lemma 14. $c(A') \leq c(A) - 1$.

Proof. Let Q be a $(0,1)-$path in $\overrightarrow{H}$ with $\delta_0'(Q) = c(A')$ and let (i_0, j_0) be the last vertex on Q that is in T. We denote the $(0,(i_0,j_0))-$subpath and the $((i_0,j_0),1)-$subpath of Q by Q_1 and Q_2, respectively. By Lemmas 11 and 13,

$$\beta_1(i_0,j_0) = a_{i_0,j_0} = a'_{i_0,j_0} + 1 = \beta_1'(i_0,j_0) + 1,$$

and w.l.o.g. we may assume $Q_1 = (0,(i_0,1),(i_0,2),\ldots,(i_0,j_0))$. For the first arc e_0 of Q_2 we have $\delta_0(e_0) = \delta_0'(e_0)$ or $\delta_0(e_0) = \delta_0'(e_0) - 1$, and for all arcs $e \neq e_0$ of Q_2, $\delta_0(e) = \delta_0'(e)$.

Case 1: $\delta_0(e_0) = \delta_0'(e_0)$.

$$\begin{aligned}\delta_0(Q) &= \delta_0(Q_1) + \delta_0(Q_2) = \delta_0'(Q_1) + 1 + \delta_0'(Q_2)\\ &= \delta_0'(Q) + 1 = c(A') + 1,\end{aligned}$$

and thus $c(A) \geq c(A') + 1$.

Case 2: $\delta_0(e_0) = \delta_0'(e_0) - 1$.

By the same argument as in the first case we only get

$$\delta_0(Q) = c(A').$$

Assume $\delta_0(Q_2) = \alpha_2(i_0,j_0)$. From

$$\alpha(i_0,j_0) = c(A) \quad \text{and} \quad \alpha_1(i_0,j_0) = a_{i_0,j_0}$$

we deduce $\delta_0(Q) = c(A)$. Now consider two cases:

1. If Q has a vertex (i,j) with $\beta_1(i,j) > a_{i,j}$, then by Lemma 12, Q contains an arc $((i_1,j_1),(i_1,j_1+1))$ with $(i_1,j_1) \in V_2$.
2. If $\beta_1(i,j) = a_{i,j}$ for every $(i,j) \in Q$, let (i_1,j_1) be the second last vertex of Q, i.e. $j_1 = n$ and $((i_1,j_1),1)$ is the last arc of Q. Note that $\beta_1(i_1,j_1) = a_{i_1,j_1}$ implies $(i_1,j_1) \in V_2$.

From $\delta_0(e_0) = \delta_0'(e_0) - 1$ follows that either

$$e_0 \in E_0^{(3)} \cup E_0^{(4)} \quad \text{or} \quad (e_0 \in E_0^{(2)} \text{ and } d_{i_0,j_0+1} \geq 0).$$

Hence, using $d_{i_0,j'} \geq 0$ for $1 \leq j' \leq j_0$, $(i_1,j_1) \notin Q_1$ and we obtain the contradiction

$$(Q_2 \setminus \{(i_0,j_0)\}) \cap V_2 \neq \emptyset.$$

Consequently, $\delta_0(Q_2) < \alpha_2(i_0,j_0)$ and there exists an $((i_0,j_0),1)-$path Q_2^* with $\delta_0(Q_2^*) > \delta_0(Q_2)$. By concatenating Q_1 and Q_2^* we obtain a $(0,1)-$path Q^* with $\delta_0(Q^*) > c(A')$, and thus

$$c(A) \geq c(A') + 1.$$

∎

Now we collect the lemmas to prove Theorem 1.

Proof of Theorem 1. That the maximal weight of a path is a lower bound for the TNMU is an immediate consequence of Lemmas 4, 6 and 7 and duality. The existence of a segmentation with $\sum_{i=1}^{k} u_i = c(A)$ is proved by induction on $c(A)$. If $c(A) = 0$ then $A = 0$ and there is nothing to do. For $c(A) > 0$ we apply Algorithm 1 to construct a segment S with $c(A - S) \leq c(A) - 1$. By induction there are segments $S_2, S_3, \ldots, S_k$ and positive integers $u_2, u_3, \ldots, u_k$ such that

$$A - S = \sum_{i=2}^{k} u_i S_i \quad \text{and} \quad \sum_{i=2}^{k} u_i = c(A - S) \leq c(A) - 1,$$

and thus with $S_1 = S$ and $u_1 = 1$,

$$A = \sum_{i=1}^{k} u_i S_i \quad \text{and} \quad \sum_{i=1}^{k} u_i = c(A - S) + 1 \leq c(A).$$

■

As observed after Example 3 each segment can be determined in time $O(m^2 n)$, so the time needed for the whole segmentation is bounded by $c(A)O(m^2 n)$. On any $0 - 1-$path in $\overrightarrow{H}$ the number of arcs with positive weight is bounded by n, because only the horizontal arcs can have positive weight. And for each of these arcs the weight is bounded by

$$L = \max\{a_{i,j} \ : \ i = 1, \ldots, m; \ j = 1, \ldots, n\}.$$

So $c(A) \leq nL$ and the complexity of the whole segmentation algorithm is $O(m^2 n^2 L)$.

2.4 Test results

Table 2.1 shows some test results of our algorithm in comparison with other algorithms. Each row shows the average TNMU for a 15×15–matrix with randomly chosen entries from $\{0, \ldots, L\}$. The columns labeled 'Xia–Verhey', 'Bortfeld' and 'Galvin' contain the results for the algorithms of Xia and Verhey [28], Bortfeld *et al* [6] and Galvin *et al* [11], respectively. The numbers in these columns are taken from Xia and Verhey [28]. The last column shows the average TNMU obtained by Engel's algorithm [10], which is TNMU–optimal for the segmentation problem without ICC. To obtain the results of the column labeled 'new' we implemented Algorithm 1 in C++. For a

L	new	Xia–Verhey	Bortfeld	Galvin	Engel (without ICC)
3	15.4	19.5	17.7	19.7	14.0
4	19.5	29.6	22.8	40.5	17.9
5	23.6	30.9	27.9	40.1	21.7
6	27.6	46.8	32.8	44.2	25.6
7	31.7	45.6	37.9	67.1	29.4
8	35.7	63.4	42.8	72.3	33.2
9	39.8	67.1	47.8	72.3	37.0
10	43.8	68.6	52.6	76.5	40.9
11	47.7	68.6	57.6	81.4	44.7
12	51.8	101.1	62.4	106.8	48.5
13	55.7	100.6	67.3	101.1	52.3
14	59.8	100.0	72.2	112.7	56.2
15	63.8	98.0	77.1	116.0	59.8
16	67.7	124.9	82.0	154.5	63.3

Tab. 2.1: Average TNMU for random 15×15–matrices with maximal entry L.

matrix A the segment S was determined and subtracted from A, and this was iterated until the zero matrix was reached. For each L this was done for 10000 random matrices A and the average TNMU was determined. On a 1.3 GHz PC the computation for the whole column (i.e. the segmentation of 140000 matrices) took 206 seconds.

3. A HEURISTIC FOR THE REDUCTION OF THE NUMBER OF SEGMENTS

In this chapter we present a greedy–heuristic that can be used to find a segmentation with minimal TNMU and a small NS. For brevity of notation we slightly modify the digraph $\overrightarrow{H}$ from Chapter 2. We add vertices $(0, i)$ and $(n+1, i)$ (for $i \in [m]$) and replace every arc $(0, (i, 1))$ by the two arcs $(0, (i, 0))$, $((i, 0), (i, 1))$ and every arc $((i, n), 1)$ by the two arcs $((i, n), (i, n+1))$, $((i, n+1), 1)$. The weights of the new arcs are determined by

$$\delta_0(0, (i, 0)) = \delta_0((i, n), (i, n+1)) = \delta_0((i, n+1), 1) = 0,$$
$$\delta_0((i, 0), (i, 1)) = a_{i,1}$$

for all $i \in [m]$. The resulting digraph is called $\overrightarrow{H}$ again. Figure 3.1 illustrates the modification for $m = n = 4$. By the results of Chapter 2, we may assume

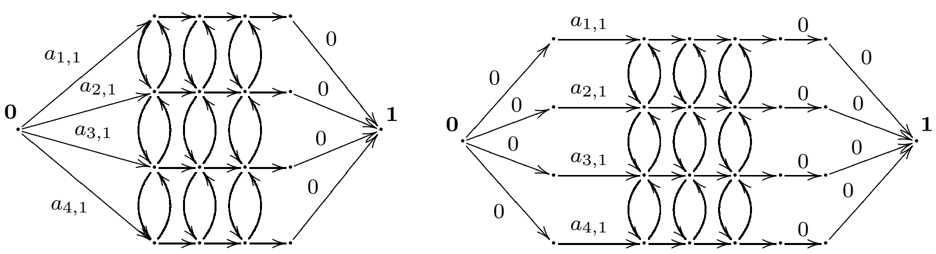


Fig. 3.1: The old and the new digraph.

that we have already determined the minimal TNMU which equals

$$c(A) = \max\{\delta_0(P) \ : \ P \text{ is a } (0, 1) - \text{path in } \overrightarrow{H}\},$$

and for every $(i, j) \in [m] \times [n]$ the values

$$\beta_1(i, j) = \max\{\delta_0(P) \ : \ P \text{ is a } (0, (i, j)) - \text{path in } \overrightarrow{H}\}, \qquad (3.1)$$
$$\beta_2(i, j) = \max\{\delta_0(P) \ : \ P \text{ is a } ((i, j), 1) - \text{path in } \overrightarrow{H}\}. \qquad (3.2)$$

3.1 The algorithm

Adopting the terminology of [10] we call the pair (u, S) of a positive integer u and a segment S an *admissible segmentation pair* if

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

The essential step of our algorithm is to determine the maximal coefficient u with the property that there exists a segment S, such that (u, S) is an admissible segmentation pair. Iterating this step with $A' = A - uS$ we clearly obtain a segmentation of A with $c(A)$ monitor units. In order to derive an upper bound for the coefficient u in an admissible segmentation pair (u, S), we identify, according to [4], 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 $(1 \le i \le m)$ and two additional vertices D and D' are added:

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

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 $l' \le r+1$ and $r' \ge l-1$. In addition, the arc set F contains all arcs from D to the first layer and from the last layer m to D', so

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

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

There is a bijection between the possible leaf positions and the paths from D to D' in Γ. This is illustrated in Fig. 3.2 which shows the paths in Γ for $m = 4$, $n = 2$, corresponding to the segments

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

Assume, for every triple (i, l, r), $1 \le i \le m$, $1 \le l \le r+1 \le n+1$, we have already determined some upper bound $u_0(i, l, r)$ for the coefficient u in

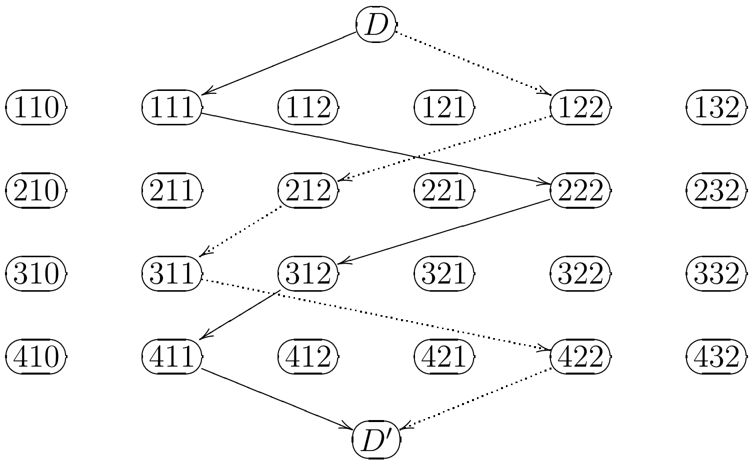


Fig. 3.2: The vertices of Γ for $m = 4$, $n = 2$ and two (D, D')–paths.

an admissible segmentation pair (u, S), where S is a segment with $l_i = l$ and $r_i = r$. In other words, $u \leq u_0(i, l_i, r_i)$ for all i if (u, S) is an admissible segmentation pair and l_i, r_i $(i = 1, \dots, m)$ are the parameters of S. We put

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

Clearly, $\hat{u}$ is an upper bound for the coefficient u in an admissible segmentation pair (u, S). Now we describe an algorithm which constructs an admissible segmentation pair (u, S) with maximal u. Fix u and assume we have already determined the first $i - 1$ rows of a segment. If it is possible to complete these $i - 1$ rows to obtain a segment S such that (u, S) is an admissible segmentation pair, then procedure `Complete Segment`(i) (Algorithm 2) determines $l_i, \dots, l_m$ and $r_i, \dots, r_m$ realizing such a completion. Here `MaxWeight`(i) denotes the maximal weight of a path in $\overrightarrow{H}$ that has all its vertices in the first i rows, where the arc weights are determined according to $A - uS$ for any segment S with parameters $l_1, \dots, l_i$ and $r_1, \dots, r_i$ in the first i rows. Clearly `MaxWeight`(i) depends only on the values that are already determined and the condition in line 4 is necessary for the possibility to continue with the candidates l_i, r_i and obtain an admissible segmentation pair. Now the pair (u, S) is constructed by procedure `Construct Segment` (Algorithm 3). Clearly, the efficiency of the backtracking depends very much on the quality of the bounds $u_0(i, l, r)$. We give some bounds that turned out to be quite good in numerical experiments. Trivially, in an admissible segmentation pair (u, S) we have, for all i,

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

Algorithm 2 Complete Segment(i)

```
for (l_i, r_i) with 1 ≤ l_i ≤ r_{i-1} + 1,
            max{l_i, l_{i-1}} - 1 ≤ r_i ≤ n and
            u_0(i, l_i, r_i) ≥ u do
  if MaxWeight(i) ≤ c(A) - u then
    if i < m then
      Complete Segment(i + 1)
    else
      finished:=true
    end if
  end if
end for
```

Algorithm 3 Construct Segment

```
u := û
finished:=false
l_0 := 1, r_0 := n + 1
while not finished do
  Complete Segment(1)
  if not finished then
    u := u - 1
  end if
end while
```

Fix an admissible segmentation pair (u, S), denote by δ_0' the weight function on $\overrightarrow{H}$ corresponding to $A' = A - uS$ and let

$$\beta_1'(i,j) = \max\{\delta_0'(P) \ : \ P \text{ is a } (0,(i,j)) - \text{path in } \overrightarrow{H}\}, \tag{3.3}$$
$$\beta_2'(i,j) = \max\{\delta_0'(P) \ : \ P \text{ is a } ((i,j),1) - \text{path in } \overrightarrow{H}\}. \tag{3.4}$$

The upper bounds below are based on the following simple observations.

1. The only arcs e with $\delta_0'(e) < \delta_0(e)$ are of the form $e = ((i, l_i - 1), (i, l_i))$ $(1 \le i \le m)$, and for these arcs $\delta_0'(e) \ge \delta_0(e) - u$.

2. For arcs of the form $e = ((i,j),(i \pm 1, j))$ with $l_i \le j \le r_i$ we have $\delta_0'(e) = \delta_0(e) + u$.

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

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

The third observation is valid since for a fixed $(0,(k,j))-$path P there is an injective mapping from the set of arcs of the form $((i, l_i - 1), (i, l_i))$ on P to the set of arcs of the form $((i,j),(i \pm 1, j))$ with $l_i \le j \le r_i$ on P: an arc $((i, l_i - 1), (i, l_i))$ is mapped to the arc $((i' \pm 1, j'), (i', j'))$ where (i', j') is the first vertex on the $((i, l_i), (k, j))-$subpath of P which is covered by some left leaf. Here the ICC assures that $l_{i' \pm 1} \le j' \le r_{i' \pm 1}$. Similarly, in the fourth observation the arcs of the form $((i, l_i - 1), (i, l_i))$ can be mapped injectively to the arcs of the form $((i,j),(i \pm 1, j))$ with $l_i \le j \le r_i$ by mapping $((i, l_i - 1), (i, l_i))$ to $((i' \pm 1, j'), (i', j'))$ where $(i' \pm 1, j')$ is the last vertex on the $((k,j),(i, l_i))-$subpath of P which is not covered by some left leaf. This is illustrated in Figure 3.3. It follows, for $1 \le i \le m$,

$$\begin{aligned} \beta_1'(i,j) &\ge \beta_1(i,j) && \text{for } j < l_i, \\ \beta_2'(i,j) &\ge \beta_2(i,j) && \text{for } j \ge l_i. \end{aligned}$$

Lemma 15. *Let (u, S) be an admissible segmentation pair with $l_i = l$ and $r_i = r$. Then $u \le v_2(i, l, r)$ where*

$$v_2(i, l, l-1) = c(A) - \beta_1(i, l-1) - \max\{0, d_{i,l}\} - \beta_2(i, l),$$

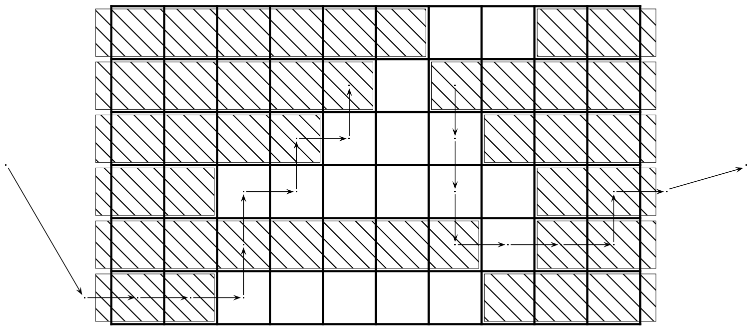


Fig. 3.3: Illustration of observations 3 and 4.

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

$$\gamma_1 = c(A) - \beta_1(i, l-1) - \beta_2(i,l),$$

$$\gamma_2 = c(A) - \beta_1(i, l-1) - \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} - \beta_2(i, r+1),$$

$$\gamma_3 = c(A) - \beta_1(i, l-1) - d_{i,l} - \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} - d_{i,r+1} - \beta_2(i, r+1),$$

$$\gamma_4 = \frac{1}{2}\left(c(A) - \beta_1(i, l-1) - \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} - d_{i,r+1} - \beta_2(i, r+1)\right).$$

Proof. Let P be the the concatenation of the paths P_1, P_2 and P_3, where P_1 is a $(0,(i,l-1))$−path with $\delta_0(P_1) = \beta_1(i,l-1)$, P_2 is the path $((i,l-1),(i,l),\ldots,(i,r+1))$, and P_3 is an $((i,r+1),1)$−path with $\delta_0(P_2) = \beta_2(i,r+1)$.

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

$$c(A) - u = c(A') \geq \delta_0'(P) \geq \beta_1(i,l-1) + \max\{0, d_{i,l}\} + \beta_2(i,l),$$

and thus $u \leq c(A) - \beta_1(i,l-1) - \max\{0, d_{i,l}\} - \beta_2(i,l)$.

Case 2: $r \geq l$. Now

$$\delta_0'(P) = \delta_0'(P_1) + \max\{0, d_{i,l} - u\} + \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} + \max\{0, d_{i,r+1} + u\} + \delta_0'(P_2),$$

and thus

$$\beta_1(i,l-1)+\max\{0,d_{i,l}-u\}+\sum_{j=l+1}^{r}\max\{0,d_{i,j}\}\\ +\max\{0,d_{i,r+1}+u\}+\beta_2(i,r+1)\le c(A)-u,$$

or

$$u+\max\{0,d_{i,l}-u\}+\max\{0,d_{i,r+1}+u\}\le c(A)-\beta_1(i,l-1)\\ -\sum_{j=l+1}^{r}\max\{0,d_{i,j}\}-\beta_2(i,r+1),$$

which implies $u \le \gamma_i$ $(i=2,3,4)$. To see $u\le\gamma_1$, consider the path Q that is the concatenation of P_1, the arc $((i,l-1),(i,l))$ and an $((i,l),1)$–path P_4 with $\delta_0(P_4)=\beta_2(i,l)$. Then

$$\delta_0'(Q)\ge\beta_1(i,l-1)+\beta_2(i,l),$$

and thus $u\le\gamma_1$. ■

Lemma 16. *Suppose* (u,S) *is an admissible segmentation pair, fix some* i*,* $2\le i\le m-1$*, and put*

$$\lambda_1=\max_{l_i\le t\le r_i}\{\beta_1(i-1,t)-a_{i-1,t}-a_{i,t}+\beta_2(i+1,t)\},$$
$$\lambda_2=\max_{l_i\le t\le r_i}\{\beta_1(i+1,t)-a_{i+1,t}-a_{i,t}+\beta_2(i-1,t)\}.$$

Then

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

Proof. By symmetry, w.l.o.g. $\lambda_1\le\lambda_2$. Assume $u>c(A)-\lambda_1$, and let t be an index where the maximum in the definition of λ_1 is attained. Let P be the concatenation of the three paths P_1, P_2 and P_3, where P_1 is an $(0,(i-1,t))$–path with $\delta_0(P_1)=\beta_1(i-1,t)$, $P_2=((i-1,t),(i,t),(i+1,t))$ and P_3 is an $((i+1,t),1)$–path with $\delta_0(P_3)=\beta_2(i+1,t)$. Then

$$\delta_0'(P)\le c(A')=c(A)-u<\lambda_1=\delta_0(P).$$

By the above observations, we have $\delta_0'(P_1)\ge\delta_0(P_1)-u$, $\delta_0'(P_3)\ge\delta_0(P_3)-u$ and

$$\delta_0'(P_2)=\begin{cases}\delta_0(P_2)+2u & \text{if } l_{i-1}\le t\le r_{i-1},\\ \delta_0(P_2)+u & \text{otherwise.}\end{cases}$$

So $\delta_0'(P) < \delta_0(P)$ implies

$$\begin{aligned} \delta_0'(P_1) &< \delta_0(P_1), \\ \delta_0'(P_2) &= \delta_0(P_2) + u, \\ \delta_0'(P_3) &< \delta_0(P_3). \end{aligned}$$

And from this follows

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

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

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

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

Thus we may put

$$u_0(i, l, r) = \min\{v_k(i, l, r) \ : \ k = 1, 2, 3\}, \tag{3.5}$$

and obtain the following result.

Theorem 2. *If the $u_0(i, l, r)$ are determined according to* (3.5) *the algorithm* `Construct Segment` *yields an admissible segmentation pair (u, S) such that $u' \leq u$ for any admissible segmentation pair (u', S').*

Example 5. For the benchmark matrix from [18] our algorithm yields the segmentation (2.4) from Example 1.

3.2 Test results

To test our algorithm we computed segmentations for 15×15–matrices with random entries from $\{0, 1, \ldots, L\}$ for $3 \leq L \leq 16$. Table 3.1 shows the results. The numbers in the columns TNMU (new) and NS (new) are the average total number of monitor units and the average number of segments, where we have averaged over 10000 matrices with randomly chosen entries from $\{0, \ldots, L\}$ (uniformly distributed). The remaining columns show the corresponding results from [28]: the columns labeled X–V, B, G contain the results for the algorithms of Xia and Verhey [28], Bortfeld *et al* [6] and Galvin *et al* [11], respectively. On an 1.3 GHz–PC the computation of the two new entries in a row of the table, i.e. the segmentation of 10000 matrices, took approximately 1 hour. But it should be mentioned that the algorithm is fast for the vast majority of the matrices, while there are some very rare exceptions. We also tested the algorithm on 13 clinical sample matrices (10×10–matrices with entries between 0 and 10). The results are shown in Table 3.2.

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

Tab. 3.1: Average TNMU and NS for random 15×15–matrices with maximal entry L.

no.	MU	NS	CPU–time
1	18	10	0.05 s
2	16	8	0.06 s
3	16	8	0.05 s
4	20	10	0.10 s
5	19	11	0.05 s
6	18	9	0.05 s
7	17	9	0.11 s
8	23	12	0.22 s
9	24	11	0.17 s
10	22	10	0.22 s
11	30	15	0.17 s
12	23	13	0.22 s
13	22	11	0.22 s

Tab. 3.2: Test results for clinical matrices.

4. EXACT MINIMIZATION OF THE NUMBER OF SEGMENTS FOR COLLIMATORS WITHOUT INTERLEAF COLLISION CONSTRAINT

The problem of minimizing the number of segments is NP–complete in the strong sense even for single row matrices. The NP–hardness was shown in [2] by reduction of 2–PARTITION [12]. Woeginger gave an unpublished proof of the NP–hardness in the strong sense by a reduction of 3–PARTITION [12]. In [15] the NS–minimization for one row has been reduced to the bipartite case of MINIMUM EDGE–COST FLOW [12]. For special instances of MINIMUM EDGE–COST FLOW there is a reduction in the reverse direction and this yields a new point of view on Woegingers argument which is presented below. The following special case of MINIMUM EDGE–COST FLOW has been shown to be strongly NP–complete in [3] by a reduction of 3–PARTITION.

Instance: A complete bipartite graph $G = (U \cup V, E)$ with $|U| = 3|V|$ and a function $w : U \to \mathbb{N} \setminus \{0\}$.

Question: Is there a flow function $f : E \to \mathbb{N}$ such that

$$\forall x \in U \quad \sum_{y \in V} f(xy) = w(x), \tag{4.1}$$

$$\forall y \in V \quad \sum_{x \in U} f(xy) = 3\overline{w}, \qquad \text{where } \overline{w} = \frac{1}{|U|} \sum_{x \in U} w(x), \tag{4.2}$$

$$|\{xy \in E \ : \ f(xy) > 0\}| \leq |U|. \tag{4.3}$$

This problem can be reduced to the NS–minimization problem as follows. We put $q = |V|$, $n = 4q$, denote the elements of U by $u_1, \dots, u_{3q}$, the elements of V by $v_1, \dots, v_q$, and define the row vector $\boldsymbol{a} = (\, a_1 \, \cdots \, a_n \,)$ as follows.

$$a_i = \sum_{j=1}^{i} w(u_j) \qquad \text{for } 1 \leq i \leq 3q,$$

$$a_i = 3(n-i)\overline{w} \qquad \text{for } 3q+1 \leq i \leq n.$$

Theorem 3 (Woeginger). *There is a segmentation of* $\boldsymbol{a}$ *with* $3q$ *segments iff there is a function* $f : E \to \mathbb{N}$ *satisfying (4.1)–(4.3).*

Proof. *"⇒":* Suppose there is a segmentation

$$\boldsymbol{a} = \sum_{j=1}^{3q} c_j \boldsymbol{s}^{(j)} \tag{4.4}$$

where the segments are described by

$$s_i^{(j)} = \begin{cases} 1 & \text{if } l_j \le i \le r_j \\ 0 & \text{otherwise.} \end{cases} \quad (j \in [3q],\ i \in [n]).$$

By Lemma 1 from [15] we may assume that $l_j \le 3q \le r_j$ for all $j \in [3q]$. Moreover, $a_i > a_{i-1}$ for all $i \in [3q]$ (where $a_0 = 0$) implies that for each $i \in [3q]$ there is some $j \in [3q]$ with $l_j = i$, hence we may assume $l_i = i$ for $i \in [3q]$. Let

$$f(u_i v_{r_i - 3q+1}) = c_i \qquad (i \in [3q])$$

and $f(xy) = 0$ for all the remaining edges xy. Observe that $a_n = 0$, so $r_i < n$ for all i and $1 \le r_i - 3q + 1 \le q$. Clearly, (4.3) is satisfied. Now fix $i \in [3q]$. From (4.4) and the fact that $j = i$ is the only index with $l_j = i$ we obtain that

$$w(u_i) = a_i - a_{i-1} = c_i = f(u_i v_{r_i}) = \sum_{y \in V} f(u_i y),$$

so (4.1) is satisfied. Now fix i, $3q + 1 \le i \le n$. From (4.4) we obtain

$$\begin{aligned} 3\overline{w} = a_{i-1} - a_i &= \sum_{j \in [3q]: r_j = i-1} c_j \\ &= \sum_{j \in [3q]: r_j = i-1} f(u_j v_{i-3q}) = \sum_{j=1}^{3q} f(u_j v_{i-3q}), \end{aligned}$$

thus (4.2) is satisfied.

"⇐": Suppose there is a function f satisfying (4.1)–(4.3). By (4.1) and (4.3), for each $j \in [3q]$ there is exactly one $k(j) \in [q]$ with $f(u_j v_{k(j)}) > 0$. For $j \in [3q]$, put

$$c_j = f(u_j v_{k(j)}), \quad l_j = j, \quad r_j = 3q + k(j) - 1,$$

and define segments $\boldsymbol{s}^{(j)}$ by

$$s_i^{(j)} = \begin{cases} 1 & \text{if } l_j \le i \le r_j \\ 0 & \text{otherwise.} \end{cases} \qquad (j \in [3q],\ i \in [n]).$$

This yields a segmentation of $\boldsymbol{a}$: for $i \le 3q$ we have $s_i^{(j)} = 1$ iff $l_j \le i$, and so

$$\sum_{j=1}^{3q} c_j s_i^{(j)} = \sum_{j=1}^{i} c_j = \sum_{j=1}^{i} \sum_{v \in V} f(u_j, v) = \sum_{j=1}^{i} w(u_j) = a_i,$$

and for $i > 3q$ we have $s_i^{(j)} = 1$ iff $r_j \ge i$, so

$$\begin{aligned} \sum_{j=1}^{3q} c_j s_i^{(j)} = \sum_{t=i}^{n-1} \sum_{j \in [3q]: r_j = t} c_j &= \sum_{t=i}^{n-1} \sum_{j \in [3q]: k(j) = t-3q+1} c_j \\ &= \sum_{t=i}^{n-1} \sum_{j=1}^{3q} f(u_j v_{t-3q+1}) = (n-i)\overline{w} = a_i. \end{aligned}$$

■

This shows that the NS–minimization is NP–hard. But the reduction essentially depends on the fact that the entries can become arbitrary large. In this chapter we show that the NS–minimization problem can be solved in time polynomial in the matrix dimensions m and n if the maximal entry L of the intensity matrix is bounded. This seems to be a reasonable assumption in practice: for instance Xia and Verhey [28] report that they obtained matrices with 7 nonzero intensity levels when they applied a preliminary version of the CORVUS inverse treatment planning system (NOMOS corporation) to a very complex head and neck tumor case. The algorithm proposed here is an application of the dynamic programming principle (see [5]).

4.1 Single row intensity maps

First we give an exact formulation of the problem L–One Row–Min MU–Min NS:

Instance: A vector $\boldsymbol{a} = (\, a_1\ a_2\ \cdots\ a_n \,)$ of integers with $0 \le a_i \le L$ $(i = 1, \ldots, n)$.

Problem: Find a segmentation with in first instance minimal TNMU and in second instance minimal NS!

As before, we put $a_0 = a_{n+1} = 0$. Let

$$P = \{i \in [n] \ : \ a_i \geq a_{i-1} \text{ and } a_i > a_{i+1}\},$$
$$Q = \{i \in [n] \ : \ a_i < a_{i-1} \text{ and } a_i \leq a_{i+1}\}.$$

Clearly, $|P| = |Q| + 1$ if $a_n \neq 0$ and $|P| = |Q|$ if $a_n = 0$. If $a_n \neq 0$ denote the elements of P and Q by $p_1, \ldots, p_t$ and $q_1, \ldots, q_{t-1}$ such that

$$p_1 < q_1 < p_2 < q_2 < \cdots < q_{t-1} < p_t,$$

and put $q_0 = 0$ and $q_t = n + 1$. If $a_n = 0$ denote the elements of P and Q by $p_1, \ldots, p_t$ and $q_1, \ldots, q_t$ such that

$$p_1 < q_1 < p_2 < q_2 < \cdots < q_{t-1} < p_t < q_t.$$

From the results of [10] it follows that in a TNMU–optimal segmentation

$$\boldsymbol{a} = \sum_{j=1}^{k} c_j \boldsymbol{s}^{(j)}$$

every segment is of the form

$$s_i^{(j)} = \begin{cases} 1 & \text{for } l_j \leq i \leq r_j, \\ 0 & \text{otherwise,} \end{cases}$$

with $q_{\tau-1} < l_j \leq p_\tau$ and $p_{\tau'} \leq r_j < q_{\tau'}$ for some $\tau, \tau' \in [t]$. Since the order of the segments is not relevant, we may order them in such a way that $r_1 \leq \cdots \leq r_k$. For $\tau \in [t-1]$, let $k_0(\tau)$ be the unique index with $r_j < q_\tau$ for $j \leq k_0(\tau)$ and $r_j \geq q_\tau$ for $j > k_0(\tau)$, and put

$$\boldsymbol{a}^{(\tau)} = \boldsymbol{a} - \sum_{j=1}^{k_0(\tau)} c_j \boldsymbol{s}^{(j)}.$$

Also put $k_0(0) = 0$, $k_0(t) = k$, $\boldsymbol{a}^{(0)} = \boldsymbol{a}$ and $\boldsymbol{a}^{(t)} = \boldsymbol{0}$. For $j > k_0(\tau)$, from $r_j \geq q_\tau$ it follows that for $i \leq q_\tau$,

$$s_i^{(j)} = 1 \quad \Longleftrightarrow \quad l_j \leq i.$$

In particular, for $i = 1, \ldots, q_\tau - 1$ and $j = k_0(\tau) + 1, \ldots, k$,

$$s_i^{(j)} = 1 \quad \Longrightarrow \quad s_{i+1}^{(j)} = 1. \tag{4.5}$$

For $0 \le \tau \le t-1$, we have

$$\boldsymbol{a}^{(\tau)} = \sum_{j=k_0(\tau)+1}^{k} c_j \boldsymbol{s}^{(j)},$$

hence (4.5) implies that

$$a_1^{(\tau)} \le a_2^{(\tau)} \le \cdots \le a_{q_\tau}^{(\tau)},$$

and the multisets

$$U_\tau = \{a_i^{(\tau)} - a_{i-1}^{(\tau)} \ : \ 1 \le i \le q_\tau,\ a_i^{(\tau)} \ne a_{i-1}^{(\tau)}\}, \tag{4.6}$$
$$V_\tau = \{a_i^{(\tau)} - a_{i-1}^{(\tau)} \ : \ q_\tau < i \le p_{\tau+1},\ a_i^{(\tau)} \ne a_{i-1}^{(\tau)}\}, \tag{4.7}$$
$$W_\tau = \{a_i^{(\tau)} - a_{i+1}^{(\tau)} \ : \ p_{\tau+1} \le i < q_{\tau+1},\ a_i^{(\tau)} \ne a_{i+1}^{(\tau)}\} \tag{4.8}$$

are partitions of a_{q_τ}, $a_{p_{\tau+1}} - a_{q_\tau}$ and $a_{p_{\tau+1}} - a_{q_{\tau+1}}$, respectively. Observe that $a_i^{(\tau)} = a_i$ for $i \ge q_\tau$, hence V_τ and W_τ depend only on $\boldsymbol{a}$, while U_τ depends also on the pairs

$$(\boldsymbol{s}^{(1)}, c_1), \ldots, (\boldsymbol{s}^{(k_0(\tau))}, c_{k_0(\tau)}).$$

Considering the sequence (U_τ, V_τ, W_τ) ($\tau = 0, \ldots, t$), where we add $U_t = V_t = W_t = \emptyset$, we will derive a method to construct the desired segmentation.

Definition 2. For integers u, v and w with $0 \le u \le v \le L$ and $0 \le w < v$, a $(u, v, w)-$*peak* is a triple (U, V, W) of unordered partitions of u, $v-u$ and $v-w$, i.e. a triple of multisets of positive integers with

$$\sum_{x \in U} x = u, \qquad \sum_{x \in V} x = v - u, \quad \sum_{x \in W} x = v - w.$$

In addition, the triple $(\emptyset, \emptyset, \emptyset)$ is called $(0,0,0)-$peak.

Thus for $\tau = 0, \ldots, t$, (U_τ, V_τ, W_τ) is an $(a_{q_\tau}, a_{p_{\tau+1}}, a_{q_{\tau+1}})-$peak (where $a_{p_{t+1}} = a_{q_{t+1}} = 0$), and for $\tau \le t-1$, the choice of the pairs

$$(\boldsymbol{s}^{(k_0(\tau)+1)}, c_{k_0(\tau)+1}), \ldots, (\boldsymbol{s}^{(k_0(\tau+1))}, c_{k_0(\tau+1)})$$

can be considered as the choice of a way to go from the peak (U_τ, V_τ, W_τ) to the peak $(U_{\tau+1}, V_{\tau+1}, W_{\tau+1})$. We claim that the number of segments needed for this step does not depend on the particular $\boldsymbol{a}^{(\tau)}$, but only on the multisets $U_\tau \cup V_\tau$, W_τ and $U_{\tau+1}$. To prove this we associate with a (u, v, w)–peak (U, V, W) a vector $\boldsymbol{b} = (\, b_1 \cdots b_\beta \,)$ as follows. Put $\alpha = |U| + |V|$, $\beta = \alpha +$

$|W|$, denote the elements of $U \cup V$ by $d_1, \dots, d_\alpha$ and the elements of W by $d_{\alpha+1}, \dots, d_\beta$, such that

$$d_1 \geq d_2 \geq \cdots \geq d_\alpha \quad \text{and} \quad d_{\alpha+1} \geq d_{\alpha+2} \geq \cdots \geq d_\beta.$$

So, for $U = U_\tau$, $V = V_\tau$ and $W = W_\tau$ the d_i $(i = 1, \dots, \beta)$ are the absolute values of the nonzero differences of consecutive entries of the initial part $\left(\begin{smallmatrix} a_1^{(\tau)} & \dots & a_{q_{\tau+1}}^{(\tau)} \end{smallmatrix} \right)$ of $\boldsymbol{a}^{(\tau)}$. Now $\boldsymbol{b}$ is defined by

$$b_i = \begin{cases} \sum\limits_{j=1}^{i} d_j & \text{for } 1 \leq i \leq \alpha, \\ v - \sum\limits_{j=\alpha+1}^{i} d_j & \text{for } \alpha+1 \leq i \leq \beta. \end{cases}$$

In addition, let $b_0 = 0$.

Example 6. The associated vector for any peak with $U \cup V = \{4, 2, 1, 1\}$ and $W = \{2, 2, 1\}$ is $\boldsymbol{b} = \left(\begin{smallmatrix} 4 & 6 & 7 & 8 & 6 & 4 & 3 \end{smallmatrix} \right)$.

Lemma 17. *Fix some τ, $0 \leq \tau \leq t-1$, and let $\boldsymbol{b} = \left(\begin{smallmatrix} b_1 & \dots & b_\beta \end{smallmatrix} \right)$ be the vector associated with the $(a_{q_\tau}, a_{p_{\tau+1}}, a_{q_{\tau+1}})$–peak (U_τ, V_τ, W_τ), defined according to (4.6)–(4.8), where $\alpha = |U_\tau \cup V_\tau|$ and $\beta = \alpha + |W_\tau|$. Also let U' be a partition of $a_{q_{\tau+1}}$, and let $c_1, \dots, c_\rho$ be positive integers with*

$$\sum_{j=1}^{\rho} c_j = a_{p_{\tau+1}} - a_{q_{\tau+1}}. \tag{4.9}$$

Then the following statements are equivalent.

1. *There exist integers l_j, r_j with $1 \leq l_j \leq p_{\tau+1} \leq r_j < q_{\tau+1}$ $(j = 1, \dots, \rho)$, such that for $\boldsymbol{a'} = \boldsymbol{a}^{(\tau)} - \sum\limits_{j=1}^{\rho} c_j \boldsymbol{s}^{(j)}$, where*

$$s_i^{(j)} = \begin{cases} 1 & \textit{if } l_j \leq i \leq r_j \\ 0 & \textit{otherwise} \end{cases} \qquad (j = 1, \dots, \rho;\ i = 1, \dots, n)$$

we have

(a) $0 \leq a'_1 \leq a'_2 \leq \cdots \leq a'_{q_{\tau+1}}$

(b) $\{a'_i - a'_{i-1} \ :\ 1 \leq i \leq q_{\tau+1},\ a'_i \neq a'_{i-1}\} = U'$ *(where $a'_0 = 0$).*

2. *There exist integers l'_j, r'_j with $1 \le l'_j \le r'_j \le \beta - 1$ for $j = 1, \dots, \rho$, such that for $\boldsymbol{b'} = \boldsymbol{b} - \sum_{j=1}^{\rho} c_j \boldsymbol{f}^{(j)}$, where*

$$f_i^{(j)} = \begin{cases} 1 & \text{if } l'_j \le i \le r'_j \\ 0 & \text{otherwise} \end{cases} \qquad (j = 1, \dots, \rho;\ i = 1, \dots, \beta)$$

we have

   (a) $b'_1 \le b'_2 \le \cdots \le b'_\beta = b_\beta$

   (b) $\{b'_i - b'_{i-1} \ : \ 1 \le i \le \beta,\ b'_i \ne b'_{i-1}\} = U'$ *(where $b'_0 = 0$).*

Proof. Let

$$R_1 = \{i \ : \ 1 \le i \le p_{\tau+1},\ a_i^{(\tau)} \ne a_{i-1}^{(\tau)}\},$$
$$R_2 = \{i \ : \ p_{\tau+1} \le i < q_{\tau+1},\ a_i^{(\tau)} \ne a_{i+1}^{(\tau)}\}.$$

Clearly,

$$U_\tau \cup V_\tau = \{a_i^{(\tau)} - a_{i-1}^{(\tau)} \ : \ i \in R_1\} \quad \text{and}$$
$$W_\tau = \{a_i^{(\tau)} - a_{i+1}^{(\tau)} \ : \ i \in R_2\}.$$

But by construction of $\boldsymbol{b}$ we also have

$$U_\tau \cup V_\tau = \{b_i - b_{i-1} \ : \ 1 \le i \le \alpha\} \quad \text{and}$$
$$W_\tau = \{b_i - b_{i+1} \ : \ \alpha \le i \le \beta - 1\}.$$

Together this implies that there are bijections

$$\varphi_1 : R_1 \to \{1, \dots, \alpha\}, \quad \varphi_2 : R_2 \to \{\alpha, \dots, \beta - 1\},$$

such that

$$a_i^{(\tau)} - a_{i-1}^{(\tau)} = b_{\varphi_1(i)} - b_{\varphi_1(i)-1} \text{ for } i \in R_1 \quad \text{and}$$
$$a_i^{(\tau)} - a_{i+1}^{(\tau)} = b_{\varphi_2(i)} - b_{\varphi_2(i)+1} \text{ for } i \in R_2.$$

It is an easy consequence of the results of [10], that from the assumption (4.9) it follows that for l_j, r_j $(j = 1, \dots, \rho)$ as in the first statement, we have $l_j \in R_1$ and $r_j \in R_2$ for all j and for l'_j, r'_j $(j = 1, \dots, \rho)$ as in the second statement we have $l'_j \le \alpha$ and $r'_j \ge \alpha$ for all j. Suppose that l_j, r_j $(j = 1, \dots, \rho)$ satisfy the conditions of the first statement. The difference of the entries number i and $i - 1$ changes only when $l_j = i$ or $r_j = i - 1$ for some j. Thus, if $i \notin R_1$ and $i - 1 \notin R_2$ we have

$$a'_i - a'_{i-1} = a_i^{(\tau)} - a_{i-1}^{(\tau)} = 0.$$

Hence, for $i = 1, \dots, q_{\tau+1}$,

$$a'_i - a'_{i-1} \neq 0 \implies i \in R_1 \text{ or } i-1 \in R_2.$$

Put

$$\begin{aligned} C_1(i) &= \{j \in [\rho] \ : \ l_j = i\} \quad \text{for } i \in R_1, \\ C_2(i) &= \{j \in [\rho] \ : \ r_j = i\} \quad \text{for } i \in R_2. \end{aligned}$$

Then

$$\begin{aligned} a'_i - a'_{i-1} &= a_i^{(\tau)} - a_{i-1}^{(\tau)} - \sum_{j \in C_1(i)} c_j \quad \text{for } i \in R_1 \\ a'_i - a'_{i+1} &= a_i^{(\tau)} - a_{i+1}^{(\tau)} - \sum_{j \in C_2(i)} c_j \quad \text{for } i \in R_2. \end{aligned}$$

By condition *(a)* of the first statement we have $a'_i - a'_{i+1} \leq 0$ for $i = 0, \dots, q_{\tau+1} - 1$. For $i \in R_2$ this yields

$$\sum_{j \in C_2(i)} c_j \geq a_i^{(\tau)} - a_{i+1}^{(\tau)},$$

and together with

$$\sum_{i \in R_2} \sum_{j \in C_2(i)} c_j = \sum_{j=1}^{\rho} c_j = a_{p_{\tau+1}} - a_{q_{\tau+1}} = \sum_{i \in R_2} \left(a_i^{(\tau)} - a_{i+1}^{(\tau)} \right)$$

we obtain for $i \in R_2$,

$$\sum_{j \in C_2(i)} c_j = a_i^{(\tau)} - a_{i+1}^{(\tau)}.$$

and thus $a'_i - a'_{i+1} = 0$ for $i \in R_2$. So the only nonzero differences $a'_i - a'_{i-1}$ come from indices $i \in R_1$. Now put $l'_j = \varphi_1(l_j)$ and $r'_j = \varphi_2(r_j)$ $(j = 1, \dots, \rho)$ and let $\boldsymbol{b'}$ be defined as in the second statement. Then $l'_j = \varphi_1(i)$ iff $j \in C_1(i)$ and $r'_j = \varphi_2(i)$ iff $j \in C_2(i)$, hence for $i \in R_1$ we have

$$\begin{aligned} b'_{\varphi_1(i)} - b'_{\varphi_1(i)-1} &= b_{\varphi_1(i)} - b_{\varphi_1(i)-1} - \sum_{j \ : \ l'_j = \varphi_1(i)} c_j \\ &= b_{\varphi_1(i)} - b_{\varphi_1(i)-1} - \sum_{j \in C_1(i)} c_j \\ &= a_i - a_{i-1} - \sum_{j \in C_1(i)} c_j \\ &= a'_i - a'_{i-1}, \end{aligned}$$

and for $i \in R_2$,

$$\begin{aligned}
b'_{\varphi_2(i)} - b'_{\varphi_2(i)+1} &= b_{\varphi_2(i)} - b_{\varphi_2(i)+1} - \sum_{j\ :\ r'_j=\varphi_2(i)} c_j \\
&= b_{\varphi_2(i)} - b_{\varphi_2(i)+1} - \sum_{j\in C_2(i)} c_j \\
&= a_i - a_{i+1} - \sum_{j\in C_2(i)} c_j \\
&= a'_i - a'_{i+1} = 0.
\end{aligned}$$

So the second statement holds, and since all the arguments are reversible, we have proved that l_j, r_j $(j = 1, \dots, \rho)$ satisfy the conditions of the first statement iff $l'_j = \varphi_1(l_j)$, $r'_j = \varphi_2(r_j)$ $(j = 1, \dots, \rho)$ satisfy the conditions of the second statement, and this proves the lemma. ■

In fact the proof shows even more than just the equivalence of the two statements: knowing l'_j and r'_j $(j = 1, \dots, \rho)$ and R_1 and R_2, we can determine the l_j, r_j $(j = 1, \dots, \rho)$ and $R' = \{i\ :\ 1 \le i \le q_{\tau+1},\ a_i^{(\tau+1)} \neq a_{i-1}^{(\tau+1)}\}$ in a number of steps that is bounded by a constant.

Example 7. Suppose $\boldsymbol{a}^{(\tau)} = \begin{pmatrix} 2\ 2\ 3\ 7\ 7\ 9\ 8\ 5\ 5\ 12 \end{pmatrix}$ with $U_\tau = \{2,1\}$, $V_\tau = \{4,2\}$, $W_\tau = \{3,1\}$, $R_1 = \{1,3,4,6\}$ and $R_2 = \{6,7\}$. The associated vector is $\boldsymbol{b} = \begin{pmatrix} 4\ 6\ 8\ 9\ 6\ 5 \end{pmatrix}$ and bijections as in the proof of Lemma 17 are given by

$$\begin{array}{lllll}
\varphi_1: & 1 \mapsto 2, & 3 \mapsto 4, & 4 \mapsto 1, & 6 \mapsto 3, \\
\varphi_2: & 6 \mapsto 5, & 7 \mapsto 4. & &
\end{array}$$

Now from

$$\begin{pmatrix} 4\ 4\ 4\ 5\ 5\ 5 \end{pmatrix} = \begin{pmatrix} 4\ 6\ 8\ 9\ 6\ 5 \end{pmatrix} - \begin{pmatrix} 0\ 2\ 2\ 2\ 0\ 0 \end{pmatrix} - \begin{pmatrix} 0\ 0\ 1\ 1\ 1\ 0 \end{pmatrix} - \begin{pmatrix} 0\ 0\ 1\ 1\ 0\ 0 \end{pmatrix},$$

where we have

$$l'_1 = 2,\ r'_1 = 4, \qquad l'_2 = 3,\ r'_2 = 5, \qquad l'_3 = 3,\ r'_3 = 4,$$

we obtain

$$l_1 = 1,\ r_1 = 7, \qquad l_2 = 6,\ r_2 = 6, \qquad l_3 = 6,\ r_3 = 7,$$

corresponding to

$$\begin{aligned}
\begin{pmatrix} 0\ 0\ 1\ 5\ 5\ 5\ 5\ 5\ 5\ 12 \end{pmatrix} = \begin{pmatrix} 2\ 2\ 3\ 7\ 7\ 9\ 8\ 5\ 5\ 12 \end{pmatrix} &- \begin{pmatrix} 2\ 2\ 2\ 2\ 2\ 2\ 2\ 0\ 0\ 0 \end{pmatrix} \\
&- \begin{pmatrix} 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \end{pmatrix} - \begin{pmatrix} 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0 \end{pmatrix}.
\end{aligned}$$

Lemma 17 motivates the following definitions.

Definition 3. Let $\boldsymbol{b} = (\,b_1 \cdots b_\beta\,)$ be the vector associated with some (u,v,w)–peak (U,V,W) where $\alpha = |U \cup V|$ and $\beta = \alpha + |W|$, and let U' be a partition of w. Let T be the set of positive integers ρ such that there are integers $l_1,\dots,l_\rho$, $r_1,\dots,r_\rho$ and coefficients $c_1,\dots,c_\rho \in I\!N \setminus \{0\}$ such that

1. $\sum_{j=1}^{\rho} c_j = v - w$,
2. $1 \le l_j \le r_j \le \beta - 1$ for $j = 1,2,\dots,\rho$.

and for $\boldsymbol{b'} = \boldsymbol{b} - \sum_{j=1}^{\rho} c_j \boldsymbol{f}^{(j)}$, where

$$f_i^{(j)} = \begin{cases} 1 & \text{if } l_j \le i \le r_j, \\ 0 & \text{otherwise,} \end{cases} \qquad (j = 1,\dots,\rho;\ i = 1,\dots,\beta)$$

we have

3. $b'_1 \le b'_2 \le \cdots \le b'_\beta = b_\beta = w$ and
4. $\{b'_i - b'_{i-1} \ :\ 1 \le i \le \beta,\ b'_i \ne b'_{i-1}\} = U'$ (with $b'_0 = 0$).

Then we define

$$\rho(\boldsymbol{b}, U') = \begin{cases} \min T & \text{if } T \ne \emptyset, \\ \infty & \text{if } T = \emptyset. \end{cases}$$

Definition 4. Let (U,V,W) and (U',V',W') be a (u,v,w)–peak and a (u',v',w')–peak, respectively, where $u' = w$. Then we put

$$\delta((U,V,W),(U',V',W')) = \rho(\boldsymbol{b}, U'),$$

where $\boldsymbol{b}$ is the vector associated with (U,V,W).

In order to model the segmentation process we define a digraph $G = (\mathcal{V}, \mathcal{E})$. The vertex set is

$$\mathcal{V} = \{(\tau, U, V_\tau, W_\tau) \ :\ \ 0 \le \tau \le t,\ U \text{ is a partition of } a_{q_\tau}\},$$

where

$$V_\tau = \{a_i - a_{i-1} \ :\ q_\tau < i \le p_{\tau+1},\ a_i \ne a_{i-1}\},$$
$$W_\tau = \{a_i - a_{i+1} \ :\ p_{\tau+1} \le i < q_{\tau+1},\ a_i \ne a_{i+1}\}$$

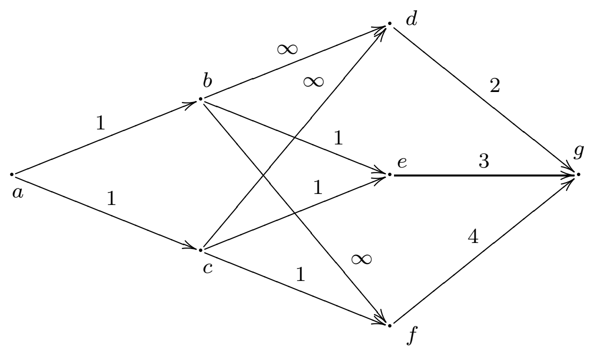


Fig. 4.1: The digraph for the vector **a**.

for $0 \leq \tau \leq t$. Observe that there is only one vertex with first component 0, namely $(0, \emptyset, V_0, W_0)$ corresponding to $\boldsymbol{a}^{(0)} = \boldsymbol{a}$ and there is only one vertex with first component t, namely $(t, \emptyset, \emptyset, \emptyset)$ corresponding to the zero vector. In general, the vertices with first component τ represent the possibilities for (U_τ, V_τ, W_τ), and by the observation before Definition 2 for each τ there is only one choice for V_τ and W_τ, depending only on $\boldsymbol{a}$. In the arc set $\mathcal{E}$ we include all arcs of the form

$$((\tau, U, V_\tau, W_\tau), (\tau + 1, U', V_{\tau+1}, W_{\tau+1}))$$

for $\tau = 0, \ldots, t-1$. Figure 4.1 shows G for $\boldsymbol{a} = (\,1\;3\;2\;4\;3\;4\,)$, where the vertices are labeled as follows.

$$\begin{array}{lll} a = (0, \emptyset, \{1,2\}, \{1\}), & b = (1, \{2\}, \{2\}, \{1\}), & c = (1, \{1,1\}, \{2\}, \{1\}), \\ d = (2, \{3\}, \{1\}, \{4\}), & e = (2, \{2,1\}, \{1\}, \{4\}), & f = (2, \{1,1,1\}, \{1\}, \{4\}) \\ g = (3, \emptyset, \emptyset, \emptyset). & & \end{array}$$

We define the arc weights in G to be the distances of the corresponding peaks, i.e.

$$\delta((\tau, U, V_\tau, W_\tau), (\tau + 1, U', V_{\tau+1}, W_{\tau+1})) = \delta((U, V_\tau, W_\tau), (U', V_{\tau+1}, W_{\tau+1}))$$

for $0 \leq \tau \leq t-1$ and all partitions U and U' of a_{q_τ} and $a_{q_{\tau+1}}$, respectively. Observe that in this definition we used the fact that (U, V_τ, W_τ) and $(U', V_{\tau+1}, W_{\tau+1})$ are an $(a_{q_\tau}, a_{p_{\tau+1}}, a_{q_{\tau+1}})$–peak and an $(a_{q_{\tau+1}}, a_{p_{\tau+2}}, a_{q_{\tau+2}})$–peak, respectively. This assures that the condition $u' = w$ in the definition of δ is satisfied. For instance, the segment $(\,1\;1\;0\;0\;0\;0\,)$ corresponds to the arc (a, b), since

$$\boldsymbol{a} - (\,1\;1\;0\;0\;0\;0\,) = (\,0\;2\;2\;4\;3\;4\,),$$

while $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$ corresponds to the arc (a, c), since

$$\boldsymbol{a} - \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 2 & 4 & 3 & 4 \end{pmatrix}.$$

In general, an arc of weight ρ corresponds to a linear combination of ρ segments. Now with a segmentation we can associate a path

$$(0, \emptyset, V_0, W_0), (1, U_1, V_1, W_1), \ldots, (t, \emptyset, \emptyset, \emptyset) \qquad (4.10)$$

in G.

Example 8. The segmentation

$$\boldsymbol{a} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} + 2\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

corresponds to the path (a, b, e, g) in Figure 4.1 as follows.

$$\begin{array}{rcr}
a & \hat{=} & \begin{pmatrix} 1 & 3 & 2 & 4 & 3 & 4 \end{pmatrix} \\
(a,b) & \hat{=} & -\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \\
b & \hat{=} & =\begin{pmatrix} 0 & 2 & 2 & 4 & 3 & 4 \end{pmatrix} \\
(b,e) & \hat{=} & -\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \\
e & \hat{=} & =\begin{pmatrix} 0 & 1 & 1 & 3 & 3 & 4 \end{pmatrix} \\
(e,g) & \hat{=} & -\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\
 & & -\begin{pmatrix} 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} \\
 & & -\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
g & \hat{=} & =\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
\end{array}$$

With these definitions the minimal number of segments needed to realize a segmentation corresponding to (4.10) equals the weight of this path.

Lemma 18. *In time $O(1)$ we can determine the values $\rho(\boldsymbol{b}, U')$ for all vectors $\boldsymbol{b}$ that are associated with some (u, v, w)–peak and for all partitions U' of w. In addition we obtain values c_j, l'_j, r'_j $(j = 1, \ldots, \rho(\boldsymbol{b}, U'))$ satisfying the conditions of Definition 3.*

Proof. The total number of vectors $\boldsymbol{b}$ associated with some $(u, v, w)-$peaks when u, v and w run through all the possible values is

$$\sum_{v=1}^{L} \sum_{w=0}^{v-1} \mathcal{P}_v \mathcal{P}_{v-w}$$

where $\mathcal{P}_i$ is the number of partitions of $i \in \mathbb{N}$. Fix one of these vectors $\boldsymbol{b}$. We consider all the sets $S = \{(l'_j, r'_j, c_j) \ : \ j = 1, \ldots, \rho\}$ $(\rho \in \mathbb{N})$, such that the vectors $\boldsymbol{f}^{(1)}, \dots, \boldsymbol{f}^{(\rho)}$, defined as in Definition 3 and the coefficients $c_1, \dots, c_\rho$ satisfy the conditions in Definition 3. We claim that there are at most

$$v^{v-w} \leq L^L$$

possibilities for S. Writing $\sum_{k=1}^{c_j} \boldsymbol{f}^{(j)}$ for $c_j \boldsymbol{f}^{(j)}$ we can express $\sum_{j=1}^{\rho} c_j \boldsymbol{f}^{(j)}$ as a sum of $\sum_{j=1}^{\rho} c_j = v - w$ $(0,1)-$vectors. In order to satisfy Conditions 1 and 3 of Definition 3, for $i = \alpha, \dots, \beta - 1$, in exactly $b_i - b_{i+1}$ of these $(0,1)-$vectors there must be 0 at position $i+1$ and a 1 at position i. So we may assume that the $v - w$ right leaf positions are fixed. Since for each right leaf position there are at most v left leaf positions the claim follows. For each S the resulting partition U' of w can be computed in $O(1)$ steps, since ρ is bounded by $v - w \leq L$, and β is bounded by $2L$. Thus the number of peaks is bounded by a constant, the number of sets S to be checked for each peak is bounded by a constant, for each of these sets the number of steps for the checking is bounded by a constant, and this completes the proof. ■

Lemma 19. *In time $O(n)$ we can determine the arc weights $\delta(e)$ for all $e \in \mathcal{E}$ and for each arc e a sequence*

$$(\boldsymbol{s}^{(1)}, c_1), \dots, (\boldsymbol{s}^{(\delta(e))}, c_{\delta(e)})$$

realizing its weight.

Proof. By Lemma 18 we may assume that we know all the $\rho(\boldsymbol{b}, U')$. First we determine in time $O(n)$ the sets

$$\begin{aligned}
P &= \{p_1, \dots, p_t\}, \\
Q &= \{q_0, \dots, q_t\}, \\
R_{1,\tau} &= \{i \ : \ q_\tau < i \leq p_{\tau+1}, \ a_i \neq a_{i-1}\} \qquad && (\tau = 0, \dots, t-1), \\
R_{2,\tau} &= \{i \ : \ p_{\tau+1} \leq i < q_{\tau+1}, \ a_i \neq a_{i+1}\} \qquad && (\tau = 0, \dots, t-1),
\end{aligned}$$

and the partitions V_τ and W_τ $(\tau = 0, \dots, t)$. By induction, we assume that we have already determined the weights of the arcs up to layer τ for some τ, $0 \leq \tau \leq t-1$. The number of vertices in layers τ and $\tau+1$ are bounded by $\mathcal{P}_{a_{q_\tau}}$ and $\mathcal{P}_{a_{q_{\tau+1}}}$, respectively. So the number of arcs is bounded by $\mathcal{P}_L^2$. Fix some $(\tau, U_\tau, V_\tau, W_\tau)$ and $(\tau+1, U_{\tau+1}, V_{\tau+1}, W_{\tau+1})$. Also by induction, we assume that we know the set

$$R_1 = \{i \ : 1 \leq i \leq p_{\tau+1}, \ a_i^{(\tau)} \neq a_{i-1}^{(\tau)}\}$$

for some possible $\boldsymbol{a}^{(\tau)}$ corresponding to $(\tau, U_\tau, V_\tau, W_\tau)$. Now by Lemma 17 (and its proof) we obtain

$$\delta((\tau, U_\tau, V_\tau, W_\tau), (\tau+1, U_{\tau+1}, V_{\tau+1}, W_{\tau+1}))$$

and a sequence realizing this value in constant time from the corresponding data for $\boldsymbol{b}$ and U' where $\boldsymbol{b}$ is the vector associated with (U_τ, V_τ, W_τ) and $U' = U_{\tau+1}$. If $\tau \leq t-2$ this also yields

$$R'_1 = \{i \ : 1 \leq i \leq p_{\tau+2},\ a_i^{(\tau+1)} \neq a_{i-1}^{(\tau+1)}\}$$

for some possible $\boldsymbol{a}^{(\tau+1)}$ corresponding to $(\tau+1, U_{\tau+1}, V_{\tau+1}, W_{\tau+1})$. So the weights for all arcs between adjacent layers can be determined in time $O(1)$. And since the number of layers $t+1$ is bounded by n, the lemma is proved. ■

Now the search for a segmentation with minimal NS amounts to the search for a path of minimal weight in a layered digraph with at most n layers where the number of vertices per layer is bounded by the constant $\mathcal{P}_L$. This can be done in time $O(n)$ ([14]). Thus we have proved

Theorem 4. L–One Row–Min MU–Min NS *can be solved in time $O(n)$.*

4.2 Multiple row intensity maps

In this subsection we generalize the basic idea of the preceding subsection to prove that for bounded L and an MLC without ICC the NS–minimization is polynomially solvable also for multiple row matrices. The problem L–Min MU–Min NS is:

Instance: An integer matrix $A = (a_{i,j})_{\substack{1\leq i\leq m\\ 1\leq j\leq n}}$ with $0 \leq a_{i,j} \leq L$ $(i \in [m], j \in [n])$.

Problem: Find a segmentation of A with in first instance minimal TNMU and in second instance minimal NS!

Assume we have already determined the minimal TNMU c. From a segmentation of A we obtain a partition $c = c_1 + c_2 + \cdots + c_k$ where c_i is the coefficient of the i–th segment $(i = 1, \ldots, k)$. First we consider the problem to check for a given partition if there is a segmentation of A with coefficients $c_1, \ldots, c_k$. This problem can be solved by checking the rows of A independently. For the moment we omit the row index and denote by $\boldsymbol{a} = (\, a_1 \cdots a_n \,)$ a fixed row of A and we put $a_0 = a_{n+1} = 0$. Compared to the single row case an additional difficulty in the multiple row case arises from the fact that the minimal TNMU that would be sufficient for a segmentation of $\boldsymbol{a}$ might be smaller than c. As a consequence we can not use Lemma 17, where condition (4.9) is essential. Here the order of the elements of the considered partition must be taken into consideration. For instance, for $\boldsymbol{b} = (\,2\;5\;0\,)$ there is a segmentation with coefficients 4, 1 and 1, namely

$$\boldsymbol{b} = 4(\,0\;1\;0\,) + (\,1\;1\;0\,) + (\,1\;0\;0\,),$$

while there is no segmentation with these coefficients for $\boldsymbol{b}' = (\,3\;5\;0\,)$. So instead of peaks we have to consider ordered peaks to be defined below. Also, in order to describe the segmentation, we attach to a peak a multiset X of coefficients, and call the result an *extended ordered peak*. This is made precise in the following definition.

Definition 5. For integers v and w with $0 \leq w < v \leq L$ an *extended ordered* (v, w)*–peak* is a pair $(\boldsymbol{b}, X)$ of an integer vector $\boldsymbol{b} = (\,b_1\;b_2\;\dots\;b_\beta\,)$, such that there is an integer α with $1 \leq \alpha < \beta$ and

$$\begin{aligned} 0 < b_1 < b_2 < \cdots < b_\alpha = v, \\ v = b_\alpha > b_{\alpha+1} > \cdots > b_\beta = w, \end{aligned}$$

and a multiset X of positive integers. In addition, a pair $(\boldsymbol{b}, X)$, where $\boldsymbol{b} = ()$ is the empty tuple and X is a multiset of positive integers is called extended ordered $(0, 0)$–peak.

Example 9. $((\,2\;5\;7\;4\;3\,), \{1, 2, 2, 3, 3\})$ is an extended ordered $(7, 3)$–peak (with $\alpha = 3$, $\beta = 5$).

Let $p_1, \dots, p_t$ and $q_0, \dots, q_t$ be defined as in the preceding section. Then for a segmentation

$$\boldsymbol{a} = \sum_{j=1}^{k} c_j \boldsymbol{s}^{(j)}$$

we can define $k_0(\tau)$ and $\boldsymbol{a}^{(\tau)}$ ($\tau = 0, \dots, t$) as before. Now for $\tau = 0, \dots, t$, we associate with $\boldsymbol{a}^{(\tau)}$ an extended ordered $(a_{p_{\tau+1}}, a_{q_{\tau+1}})$–peak $(\boldsymbol{b}^{(\tau)}, X_\tau)$ as follows. For $\tau < t$, let

$$\begin{aligned} I_\tau &= \{i \;:\; 1 \leq i \leq p_{\tau+1},\; a_i^{(\tau)} \neq a_{i-1}^{(\tau)}\}, \\ J_\tau &= \{i \;:\; p_{\tau+1} < i \leq q_{\tau+1},\; a_i^{(\tau)} \neq a_{i-1}^{(\tau)}\}, \end{aligned}$$

denote the elements of I_τ by $i_1, \dots, i_\alpha$ and the elements of J_τ by $i_{\alpha+1}, \dots, i_\beta$ such that $i_1 < i_2 < \cdots < i_\beta$, and put

$$b_0 = 0, \quad b_l = a_{i_l} \quad (l = 1, \dots, \beta).$$

Let $X_0 = \{c_1, \dots, c_k\}$ and

$$X_{\tau+1} = X_\tau \setminus \{c_{k_0(\tau)+1}, c_{k_0(\tau)+2}, \dots, c_{k_0(\tau+1)}\} \quad (\tau = 0, \dots, t-1).$$

Now for $\tau < t$, $(\boldsymbol{b}^{(\tau)}, X_\tau)$ describes the initial part of $\boldsymbol{a}^{(\tau)}$ (up to column $q_{\tau+1}$) together with the coefficients available for the remaining segments. In the final state ($\tau = t$) we have the zero row $\boldsymbol{a}^{(t)} = 0$ and a multiset X_t of coefficients, that are not needed for the considered row. With the zero row we associate the empty tuple $\boldsymbol{b}^{(t)} = ()$, and thus we obtain from any segmentation a sequence $(\boldsymbol{b}^{(0)}, X_0)$, $(\boldsymbol{b}^{(1)}, X_1), \ldots, (\boldsymbol{b}^{(t)}, X_t)$ of extended ordered peaks.

Example 10. Suppose $\boldsymbol{a} = \begin{pmatrix} 2 & 4 & 3 & 1 & 6 & 3 & 0 & 6 & 1 \end{pmatrix}$ is a row in an intensity matrix with minimal TNMU $c = 18$, and we are checking the partition $c = 5+3+2+2+2+1+1+1+1$. Then from the segmentation

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

we obtain

τ	$\boldsymbol{a}^{(\tau)}$	$\boldsymbol{b}^{(\tau)}$	X_τ
0	$\begin{pmatrix} 2 & 4 & 3 & 1 & 6 & 3 & 0 & 6 & 1 \end{pmatrix}$	$\begin{pmatrix} 2 & 4 & 3 & 1 \end{pmatrix}$	{5,3,2,2,2,1,1,1,1}
1	$\begin{pmatrix} 0 & 1 & 1 & 1 & 6 & 3 & 0 & 6 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 6 & 3 & 0 \end{pmatrix}$	{5,3,2,2,1,1,1}
2	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 1 \end{pmatrix}$	$\begin{pmatrix} 6 & 1 & 0 \end{pmatrix}$	{5,2,1,1}
3	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	$()$	{2,1}

That the vectors $\boldsymbol{b}^{(\tau)}$ provide enough information to construct the segmentation, follows from the simple observation, that w.l.o.g. a sequence of consecutive entries of equal value

$$a_{i_1} = a_{i_1+1} = \cdots = a_{i_2}$$

can be considered as one single entry. This is intuitively clear and proved formally in the next lemma.

Lemma 20. *Let $\boldsymbol{a} = \sum_{j=1}^{k} c_j \boldsymbol{s}^{(j)}$ be a segmentation with*

$$s_i^{(j)} = \begin{cases} 1 & \text{if } l_j \le i \le r_j \\ 0 & \text{otherwise} \end{cases} \qquad (j = 1, \ldots, k).$$

There are integers l'_j and r'_j $(j = 1, \ldots, k)$ with the following properties.

1. *We have* $\boldsymbol{a} = \sum_{j=1}^{k} c_j \boldsymbol{s'}^{(j)}$ *where*

$$s_i'^{(j)} = \begin{cases} 1 & \text{if } l_j' \le i \le r_j' \\ 0 & \text{otherwise} \end{cases} \qquad (j = 1, \ldots, k).$$

2.

$$a_i = a_{i-1} \implies s_i'^{(j)} = s_{i-1}'^{(j)} \qquad (i = 2, \ldots, n;\ j = 1, \ldots, k). \tag{4.11}$$

Proof. In order to satisfy the last condition, we have to replace the segments with $s_i^{(j)} \neq s_{i-1}^{(j)}$ but $a_i = a_{i-1}$ for some i. Our strategy is to modify the given segments as follows. For each sequence of consecutive entries of $\boldsymbol{a}$ of equal value we choose one representative, for instance the rightmost one, and adapt the entries for each segment to the chosen column. This corresponds to the following shifting of the leaves: if the left leaf covers a part of the plateau it is shifted to the right until the whole plateau is open, and if the right leaf covers a part of the plateau it is shifted to the left until the whole plateau is covered.

First observe that $s_i^{(j)}$ can differ from $s_{i-1}^{(j)}$ only if $i = l_j$ or $i - 1 = r_j$. So for (4.11) it is sufficient that, for all j, we have

$$a_{l_j'} \neq a_{l_j'-1} \quad \text{and} \quad a_{r_j'} \neq a_{r_j'+1}. \tag{4.12}$$

Suppose $a_{l_j} = a_{l_j-1}$ for some j. Then $i_1 < l_j \le i_2$ for some i_1, i_2 with

$$a_{i_1} = a_{i_1+1} = \cdots = a_{i_2} = a \quad \text{and} \quad a_{i_1-1}, a_{i_2+1} \neq a. \tag{4.13}$$

Since we want to adapt the entries of the segment to the rightmost column i_2 we have to shift the left leaf to the left and put $l_j' = i_1$. Similarly, if $a_{r_j} = a_{r_j+1}$, then $i_1 \le r_j < i_2$ for some i_1, i_2 with (4.13), and in order to adapt the entries of the segment to column i_2, we have to shift the right leaf to the left and put $r_j' = i_1 - 1$. In summary, for $j \in [k]$ we put

$$l_j' = \begin{cases} l_j & \text{if } a_{l_j} \neq a_{l_j-1}' \\ \max\{i < l_j \ :\ a_i \neq a_{l_j}\} + 1 & \text{if } a_{l_j} = a_{l_j-1}' \end{cases}$$

$$r_j' = \begin{cases} r_j & \text{if } a_{r_j} \neq a_{r_j+1}' \\ \max\{i < r_j \ :\ a_i \neq a_{r_j}\} & \text{if } a_{r_j} = a_{r_j+1}' \end{cases}$$

Then (4.12) is valid for all j, hence (4.11) is satisfied. In order to check the first condition of the lemma, fix some $i \in [n]$. If $s_i'^{(j)} = s_i^{(j)}$ for all j, then

$$\sum_{j=1}^{k} c_j s_i'^{(j)} = \sum_{j=1}^{k} c_j s_i^{(j)} = a_i.$$

So assume $s_i'^{(j)} \neq s_i^{(j)}$ for some j. By construction this can be the case only if $a_i = a_{i-1}$ or $a_i = a_{i+1}$. Now let i_1 and i_2 be the indices with $i_1 \leq i \leq i_2$,

$$a_{i_1} = a_{i_1+1} = \cdots = a_i = \cdots = a_{i_2} \quad \text{and} \quad a_{i_1-1}, a_{i_2+1} \neq a_i.$$

We claim that $s_i'^{(j)} = s_{i_2}^{(j)}$ $(j = 1, \dots, k)$. If $s_{i_2}^{(j)} = 0$, $l_j > i_2$ or $r_j < i_2$. By construction, in the first case $l_j' > i_2$ and in the second case $r_j' < i_1$, so in both cases $s_i'^{(j)} = 0$. If $s_{i_2}^{(j)} = 1$, $l_j \leq i_2$ and $r_j \geq i_2$. By construction, $l_j' \leq i_1$ and $r_j' \geq i_2$, hence $s_i'^{(j)} = 1$ and the claim is proved. From this follows

$$\sum_{j=1}^{k} c_j s_i'^{(j)} = \sum_{j=1}^{k} c_j s_{i_2}^{(j)} = a_{i_2} = a_i,$$

and since this argument works for any $i \in [n]$ the first condition of the lemma is satisfied. ■

By Lemma 20 applied to $\boldsymbol{a}^{(\tau)}$, w.l.o.g. we may assume that $a_{l_j}^{(\tau)} \neq a_{l_j-1}^{(\tau)}$ and $a_{r_j}^{(\tau)} \neq a_{r_j+1}^{(\tau)}$ for all $j > k_0(\tau)$. With this assumption the next lemma, whose proof is obvious, justifies that we use the $\boldsymbol{b}^{(\tau)}$ instead of the $\boldsymbol{a}^{(\tau)}$.

Lemma 21. *For fixed τ, $0 \leq \tau \leq t-1$, let $\boldsymbol{b}^{(\tau)}$ and X_τ be defined as described above and let $\{c_1, \dots, c_\rho\} \subseteq X_\tau$ be fixed. If $a_{q_{\tau+1}} \neq 0$ let $\boldsymbol{g} = \begin{pmatrix} g_1 & \cdots & g_\gamma \end{pmatrix}$ be some vector with*

$$0 < g_1 < \cdots < g_\gamma = a_{q_{\tau+1}}.$$

Then the following statements are equivalent.

1. *There exist integers l_j, r_j with $1 \leq l_j \leq r_j < q_{\tau+1}$, $a_{l_j}^{(\tau)} \neq a_{l_j-1}^{(\tau)}$ and $a_{r_j}^{(\tau)} \neq a_{r_j+1}^{(\tau)}$ $(j = 1, \dots, \rho)$ such that for $\boldsymbol{a'} = \boldsymbol{a}^{(\tau)} - \sum_{j=1}^{\rho} c_j \boldsymbol{s}^{(j)}$, where*

$$s_i^{(j)} = \begin{cases} 1 & \text{if } l_j \leq i \leq r_j \\ 0 & \text{otherwise} \end{cases} \qquad (j = 1, \dots, \rho;\ i = 1, \dots, n)$$

   we have

   (a) $0 \leq a_1' \leq a_2' \leq \cdots \leq a_{q_{\tau+1}}' = a_{q_{\tau+1}}$

   (b) If $a_{q_{\tau+1}} \neq 0$ there are exactly γ indices $1 \leq i_1 < \cdots < i_\gamma \leq q_{\tau+1}$ with $a_{i_*}' \neq a_{i_*-1}'$ (where $a_0' = 0$) and we have

$$\begin{pmatrix} a_{i_1} & a_{i_2} & \dots & a_{i_\gamma} \end{pmatrix} = \boldsymbol{g}.$$

2. *There exist integers* l'_j, r'_j *with* $1 \le l'_j \le r'_j \le \beta - 1$ *for* $j = 1, \ldots, \rho$, *such that for* $\boldsymbol{b'} = \boldsymbol{b} - \sum_{j=1}^{\rho} c_j \boldsymbol{f}^{(j)}$, *where*
$$f_i^{(j)} = \begin{cases} 1 & \text{if } l'_j \le i \le r'_j \\ 0 & \text{otherwise} \end{cases} \qquad (j = 1, \ldots, \rho;\ i = 1, \ldots, \beta)$$
we have
   (a) $b'_1 \le b'_2 \le \cdots \le b'_\beta = b_\beta = a_{q_{\tau+1}}$
   (b) *If* $a_{q_{\tau+1}} \neq 0$ *there are exactly* γ *indices* $1 \le i_1 < \cdots < i_\gamma \le \beta$ *with* $b'_{i_*} \neq b'_{i_*-1}$ *(where* $b'_0 = 0$*) and we have*
$$\begin{pmatrix} b_{i_1} & b_{i_2} & \ldots & b_{i_\gamma} \end{pmatrix} = \boldsymbol{g}.$$

Now for $\tau = 0, 1, \ldots, t-1$ the choice of the pairs
$$\left(\boldsymbol{s}^{k_0(\tau)+1}, c_{k_0(\tau)+1}\right), \ldots, \left(\boldsymbol{s}^{k_0(\tau+1)}, c_{k_0(\tau+1)}\right)$$
can be viewed as a way to go from the extended ordered $(a_{p_{\tau+1}}, a_{q_{\tau+1}})$–peak $(\boldsymbol{b}^{(\tau)}, X_\tau)$ to the extended ordered $(a_{p_{\tau+2}}, a_{q_{\tau+2}})$–peak $(\boldsymbol{b}^{(\tau+1)}, X_{\tau+1})$ (with $a_{p_{t+1}} = a_{q_{t+1}} = 0$).

Definition 6. Let $0 \le w < v$ and let $(\boldsymbol{b}, X)$ be an extended ordered (v, w)–peak, and let v', w' be integers with $w \le v' \le L$ and $0 \le w' < v'$ or $v' = w' = 0$. In addition let X' be a submultiset of X and denote the elements of X' by $x_1, \ldots, x_{|X'|}$. We call an extended ordered (v', w')–peak $(\boldsymbol{b'}, X \setminus X')$ *accessible* from $(\boldsymbol{b}, X)$ if there are integers $l'_1, \ldots, l'_{|X'|}, r'_1, \ldots, r'_{|X'|}$ such that

1. $1 \le l'_j \le r'_j \le \beta - 1$ for $j = 1, \ldots, |X'|$ (where $\boldsymbol{b} = \left(\begin{smallmatrix} b_1 & \cdots & b_\beta \end{smallmatrix} \right)$).

and for $\boldsymbol{b''} = \boldsymbol{b} - \sum\limits_{j=1}^{|X'|} x_j \boldsymbol{f}^{(j)}$, where
$$f_i^{(j)} = \begin{cases} 1 & \text{if } l'_j \le i \le r'_j, \\ 0 & \text{otherwise}, \end{cases} \quad (j = 1, \ldots, |X'|;\ i = 1, \ldots, \beta)$$
we have $\boldsymbol{b''} = \boldsymbol{0}$ if $v' = w' = 0$ and otherwise

2. $b''_1 \le b''_2 \le \cdots \le b''_\beta = b_\beta = w$ and
3. If $i_1 < i_2 < \cdots < i_{\gamma'}$ are the indices with $b''_{i_*} \neq b''_{i_*-1}$ (where $b''_0 = 0$), then
$$b'_1 < b'_2 < \cdots < b'_{\gamma'} = w,$$
and we have
$$\begin{pmatrix} b''_{i_1} & b''_{i_2} & \ldots & b''_{i_{\gamma'}} \end{pmatrix} = \begin{pmatrix} b'_1 & b'_2 & \ldots & b'_{\gamma'} \end{pmatrix}.$$

The definition can be interpreted as follows. Assume $a_{p_1} = v$, $a_{q_1} = w$, $a_{p_2} = v'$, $a_{q_2} = w'$, let $\boldsymbol{b}^{(0)}$ be associated with $\boldsymbol{a}^{(0)}$ as above, and let $\boldsymbol{b'} = \begin{pmatrix} b'_1 & \cdots & b'_{\beta'} \end{pmatrix}$ be a vector with

$$0 < b'_1 < \cdots < b'_{\alpha'} = v',\ v' = b'_{\alpha'} > \cdots > b'_{\beta'} = w'.$$

Then $(\boldsymbol{b'}, X \setminus X')$ is accessible from $(\boldsymbol{b}^{(0)}, X)$ iff we can assign segments $\boldsymbol{s}^{(j)}$ to the elements of X', described by l_j, r_j ($j = 1, \ldots, |X'|$) with $r_j < q_1$ for all j, such that for

$$\boldsymbol{a}^{(1)} = \boldsymbol{a}^{(0)} - \sum_{j=1}^{|X'|} x_j \boldsymbol{s}^{(j)}$$

we have $a_1^{(1)} \leq a_2^{(1)} \leq \cdots \leq a_{p_2}^{(1)}$ and the extended ordered (v', w')–peak associated with $\boldsymbol{a}^{(1)}$ is $(\boldsymbol{b'}, X \setminus X')$.

Example 11. Let $\boldsymbol{a} = \begin{pmatrix} 0 & 2 & 5 & 5 & 7 & 4 & 3 & 3 & 5 & 6 & 8 & 2 \end{pmatrix}$, $X = \{5, 3, 2, 2, 2, 1, 1, 1\}$ and $X' = \{3, 1\}$. The associated extended ordered $(7, 3)$–peak is $(\boldsymbol{b}, X)$ where $\boldsymbol{b} = \begin{pmatrix} 2 & 5 & 7 & 4 & 3 \end{pmatrix}$. Now we want to determine the extended ordered $(8, 0)$–peaks $(\boldsymbol{b'}, X \setminus X')$ that are accessible from $(\boldsymbol{b}, X)$, where

$$\boldsymbol{b'} = \begin{pmatrix} b'_1 & \ldots & b'_\gamma = 3 & 5 & 6 & 8 & 2 \end{pmatrix}.$$

We obtain that $(\boldsymbol{b'}, X \setminus X')$ and $(\boldsymbol{b''}, X \setminus X')$ are accessible from $(\boldsymbol{b}, X)$, where $\boldsymbol{b'} = \begin{pmatrix} 2 & 3 & 5 & 6 & 8 & 2 \end{pmatrix}$ and $\boldsymbol{b''} = \begin{pmatrix} 1 & 3 & 5 & 6 & 8 & 2 \end{pmatrix}$:

$$\begin{aligned} \begin{pmatrix} 2 & 2 & 3 & 3 & 3 \end{pmatrix} &= \boldsymbol{b} - \begin{pmatrix} 0 & 3 & 3 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \\ \begin{pmatrix} 1 & 1 & 3 & 3 & 3 \end{pmatrix} &= \boldsymbol{b} - \begin{pmatrix} 0 & 3 & 3 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \end{pmatrix}. \end{aligned}$$

This corresponds to the following possible beginnings of a segmentation.

$$\begin{aligned} &\begin{pmatrix} 0 & 2 & 5 & 5 & 7 & 4 & 3 & 3 & 5 & 6 & 8 & 2 \end{pmatrix} \\ -&\begin{pmatrix} 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ -&\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ =&\begin{pmatrix} 0 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 5 & 6 & 8 & 2 \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} &\begin{pmatrix} 0 & 2 & 5 & 5 & 7 & 4 & 3 & 3 & 5 & 6 & 8 & 2 \end{pmatrix} \\ -&\begin{pmatrix} 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ -&\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ =&\begin{pmatrix} 0 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 5 & 6 & 8 & 2 \end{pmatrix}. \end{aligned}$$

On the other hand one can check that $((\,{\scriptstyle 3\ 5\ 6\ 8\ 2}\,), X \setminus X')$ is not accessible from $(\boldsymbol{b}, X)$ and this corresponds to the fact that it is not possible to find (l_1, r_1) and (l_2, r_2) with $r_1, r_2 < 7$ such that after subtracting the corresponding segments with coefficients 3 and 1 from $\boldsymbol{a}$ we obtain a row vector $\boldsymbol{a'}$ with $a'_1 = \cdots = a'_i = 0$, $a'_{i+1} = \cdots = a'_7 = 3$ for some i, $1 \le i \le 6$. Similar statements can be made for $\boldsymbol{b'} = (\,{\scriptstyle 1\ 2\ 3\ 5\ 6\ 8\ 2}\,)$.

Lemma 22. *Let $(\boldsymbol{b}, X)$ be an extended ordered $(v, w)-$peak. Then the set of all $(\boldsymbol{b'}, X \setminus X')$ that are accessible from $(\boldsymbol{b}, X)$ can be determined in time $O(1)$.*

Proof. Observe that the accessibility does not depend on the whole vector $\boldsymbol{b'}$ but only on the initial part $(\,{\scriptstyle b'_1\ \cdots\ b'_{\gamma'}=w}\,)$. So in order to determine the accessible extended ordered peaks it is sufficient to determine the pairs $((\,{\scriptstyle b'_1\ \cdots\ b'_{\gamma'}}\,), X \setminus X')$ of initial parts and multisets of coefficients. Let $\boldsymbol{b} = (\,{\scriptstyle b_1\ \cdots\ b_\beta}\,)$ and let α be the unique index with $b_\alpha = v$. We have $b_1 < \cdots < b_\alpha$ and $b_\alpha > \cdots > b_\beta$. So for $1 \le k \le v-1$ there are at most two indices i and i' with $1 \le i, i' \le \beta - 1$ and $b_i = k$, $b_{i'} = k$ (namely the first one with $1 \le i \le \alpha - 1$ and the second one with $\alpha + 1 \le i' \le \beta - 1$). The only index i with $b_i = v$ is $i = \alpha$, and so we have

$$\sum_{i=1}^{\beta-1} b_i \le v + 2\sum_{k=1}^{v-1} k \le L^2.$$

Hence it is sufficient to consider at most $\mathcal{P}_{L^2}$ candidates for X', where each of these has at most L^2 elements. Fix one of these X'. Labeling the elements of X' as in Definition 6, for each $x_j \in X'$ there are at most $\binom{2L-1}{2}$ choices for $\boldsymbol{f}^{(j)}$. So the total number of choices for the pairs $(\boldsymbol{f}^{(j)}, x_j)$ that have to be considered is bounded by

$$\left[\binom{2L-1}{2}\right]^{|X'|} \le \left[\binom{2L-1}{2}\right]^{L^2}.$$

For each of these choices the time needed to determine the resulting $\boldsymbol{b''}$ is bounded by a constant. Precisely, in order to subtract one of the $x_j \boldsymbol{f}^{(j)}$ we have to do at most $2L$ subtractions. So after at most $L^2 \cdot 2L$ subtractions we have determined $\boldsymbol{b''}$. Finally, in order to determine the corresponding $(\,{\scriptstyle b'_1\ \cdots\ b'_{\gamma'}}\,)$ according to condition 3 of Definition 6, we have to run through the at most $2L$ entries of $\boldsymbol{b'}$. This proves the lemma, since the number of steps to determine the required data is bounded by

$$\mathcal{P}_{L^2}\left[\binom{2L-1}{2}\right]^{L^2}(L^2+1)2L.$$

■

In order to model the segmentation we construct sets $\mathcal{V}_0,\dots,\mathcal{V}_t$ of extended ordered peaks. Put $\mathcal{V}_0=\{(\boldsymbol{b}^{(0)},X_0)\}$ and suppose we have already constructed $\mathcal{V}_0,\dots,\mathcal{V}_\tau$ for some τ with $0\leq\tau<t$. Now we put

$$\mathcal{V}_{\tau+1}=\{(\boldsymbol{b'},X')\ :\ (\boldsymbol{b'},X')\text{ is an }(a_{p_{\tau+2}},a_{q_{\tau+2}})-\text{peak that is accessible from some }(\boldsymbol{b},X)\in\mathcal{V}_\tau\}.$$

Here for brevity of notation we put $a_{p_{t+1}}=1$ and $a_{q_{t+1}}=0$. The elements of $\mathcal{V}_\tau$ represent the possibilities for $(\boldsymbol{b}^{(\tau)},X_\tau)$. There is a segmentation of the row with coefficients $c_1,\dots,c_k$ iff $\mathcal{V}_t\neq\emptyset$. Note that a natural interpretation of this construction is a breadth first search (BFS) in the tree with vertex set $\mathcal{V}_0\cup\ldots\cup\mathcal{V}_t$ starting at $(\boldsymbol{b}^{(0)},X_0)$, where two vertices $(\boldsymbol{b},X)$ and $(\boldsymbol{b'},X')$ are connected by an edge iff $(\boldsymbol{b},X)\in\mathcal{V}_\tau$, $(\boldsymbol{b'},X')\in\mathcal{V}_{\tau+1}$ for some τ and $(\boldsymbol{b'},X')$ is accessible from $(\boldsymbol{b},X)$.

Lemma 23. *For given $\mathcal{V}_\tau$, the set $\mathcal{V}_{\tau+1}$ can be determined in time $O(n^{L+1})$.*

Proof. According to [10], the sum of the elements of X_0, which is the minimal TNMU equals

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

Now in any partition $c=c_1+\cdots+c_k$ where the c_i ($i\in[k]$) are the coefficients of a segmentation of A, we have $c_i\leq L$ for $i\in[k]$. Hence such a partition can be described by an L–tuple $(\lambda_1,\dots,\lambda_L)$ of integers, where λ_r is the number of summands equal to r for $r\in[L]$. Then

$$\lambda_r\leq\frac{nL}{r}\quad(r\in[L]),$$

and so there are $O(n^L)$ choices for X_0. Now the multiset X in

$$(\boldsymbol{b},X)\in\mathcal{V}_\tau$$

is a partition of some c' with $0\leq c'\leq c\leq nL$ with all summands less than or equal to L. So there are nL possibilities for c', and for each of these there are $O(n^L)$ possible partitions. Thus the number of choices for X is bounded by $O(n^{L+1})$. The vectors $\boldsymbol{b}$ in the elements of $\mathcal{V}_\tau$ differ only in the initial part $(b_1 \dots b_\gamma)$, where $b_\gamma = a_{q_\tau}$. But these initial parts are in bijection to the ordered partitions of a_{q_τ}, and of these there are (see for instance [1])

$$\sum_{i=1}^{a_{q_\tau}} \binom{a_{q_\tau}-1}{i-1} \leq L \binom{L}{\lfloor \frac{L}{2} \rfloor}.$$

Since L is bounded by a constant we obtain that $|\mathcal{V}_\tau|$ is bounded by $O(n^{L+1})$. By Lemma 22, for each $(\boldsymbol{b}, X) \in \mathcal{V}_\tau$ the set of accessible $(\boldsymbol{b'}, X \setminus X')$ can be determined in time bounded by a constant, and this yields the claim. ■

Lemma 24. *For a fixed partition $c = c_1 + \cdots + c_k$, it can be checked in time $O(n^{L+2})$ if there is a segmentation of $\boldsymbol{a}$ with coefficients $c_1, \dots, c_k$.*

Proof. We only have to check if $\mathcal{V}_t \neq \emptyset$. Since $t \leq n$ the claim is an immediate consequence of Lemma 23. ■

Now we can prove

Theorem 5. *The problem L–*Min *MU–*Min *NS can be solved in time $O(mn^{2L+2})$.*

Proof. Obviously,

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

can be determined in time $O(mn)$. As in the proof of Lemma 23 the number of partitions of $c = c_1 + \cdots + c_k$ that have to be considered is bounded by $O(n^L)$. By Lemma 24, for a fixed partition $c = c_1 + \cdots + c_k$ it can be checked in time $O(mn^{L+2})$ if there is a segmentation of A with coefficients $c_1, \dots, c_k$, and this concludes the proof. ■

We finish this section with a remark concerning practical aspects of this result. Though the time complexity of the NS–minimization is polynomial in m and n the exponent grows linearly with L and also the L–dependent constants that were used to estimate the time–complexities of the different steps of the algorithm, grow rapidly with L. So we expect an efficient algorithm only for very small L. In the proof of the polynomiality we constructed the whole sets $\mathcal{V}_\tau$ $(\tau = 1, \dots, t)$, i.e. we performed a BFS as described before Lemma 23. But in order to decide if there is a segmentation with the considered coefficients we need to know only if $\mathcal{V}_t$ is nonempty, and in order to reconstruct a segmentation basically one path from the unique element of $\mathcal{V}_0$ to some element of $\mathcal{V}_t$ is sufficient. So for practical purposes it is natural to use depth first search (DFS) instead of BFS.

4.3 Test results

We implemented the algorithm described in the preceding section and Tables 4.1 and 4.2 show test results for random 10×10– and 15×15–matrices, respectively. The computations where done on a 2 GHz workstation and we determined the minimal NS for 1000 randomly generated matrices with maximal entry L. The entry in column 'max. time' is the maximal time needed for one single matrix, and the entry in column 'total time' is the time needed for all the 1000 matrices. For comparison the tables also contain heuristic results that were obtained with a slightly improved version of the algorithm described in [10]. In order to evaluate the performance of the

	exact			heuristic	
L	NS	max. time	total time	NS	total time
3	6.9	1 s	9 s	6.9	0.9 s
4	7.6	1 s	13 s	7.8	1.0 s
5	8.1	1 s	29 s	8.4	1.1 s
6	8.5	21 s	99 s	8.9	1.2 s
7	8.8	50 s	5.6 min	9.3	1.2 s
8	9.1	66 s	6.2 min	9.7	1.3 s
9	9.3	3.4 min	16.1 min	10.0	1.3 s
10	9.5	5.6 min	41.3 min	10.3	1.4 s
11	9.8	11.0 min	1.3 h	10.6	1.4 s
12	9.9	24.0 min	2.0 h	10.9	1.5 s
13	10.0	1.4 h	7.0 h	11.1	1.5 s

Tab. 4.1: Average number of segments for random 10×10–matrices with maximal entry L. Each entry is averaged over 1000 matrices.

heuristic we determined the differences between the heuristic values and the exact minimums. Tables 4.3 and 4.4 show the frequencies of the values of the differences when 1000 matrices where treated for each value of L. We conclude that for the considered range of parameters the exact algorithm yields only small improvements in terms of the number of segments, while the computational effort is extremely high already for small values of L. So for practical purposes the heuristic seems to be a good compromise between computation time and accuracy of the optimization. Finally, we also tested our algorithm with the 13 clinical matrices, and the results are shown in Table 4.5.

	exact			heuristic	
L	NS	max. time	total time	NS	total time
3	9.7	1 s	16 s	9.8	4.8 s
4	10.7	1 s	31 s	10.9	5.4 s
5	11.3	12 s	175 s	11.7	5.8 s
6	11.8	54 s	18.6 min	12.4	6.5 s
7	12.3	3.1 min	0.8 h	13.0	6.8 s
8	12.6	4.5 h	14.7 h	13.5	7.1 s
9	12.9	24.1 h	37.9 h	14.0	7.4 s
10	13.2	10.0 h	44.7 h	14.5	7.6 s

Tab. 4.2: Average number of segments for random 15×15–matrices with maximal entry L. Each entry is averaged over 1000 matrices.

L	0	1	2	3
3	969	31	0	0
4	876	123	1	0
5	780	218	2	0
6	663	331	2	0
7	525	456	19	0
8	437	516	47	0
9	335	603	62	0
10	306	584	104	6
11	262	615	121	2
12	168	654	173	5
13	141	641	213	5

Tab. 4.3: Frequencies of the differences between the heuristic number of segments and the exact minimum for 10×10–matrices.

L	0	1	2	3	4
3	940	60	0	0	0
4	809	189	2	0	0
5	609	379	12	0	0
6	453	509	37	1	0
7	327	585	86	2	0
8	250	594	151	5	0
9	150	609	230	11	0
10	85	551	335	28	1

Tab. 4.4: Frequencies of the differences between the heuristic number of segments and the exact minimum for 15×15–matrices.

		exact		heuristic	
no.	MU	NS	CPU–time	NS	CPU–time
1	16	7	0.04 s	8	0.01 s
2	16	7	0.19 s	7	0.00 s
3	20	8	0.39 s	8	0.01 s
4	19	7	0.04 s	8	0.00 s
5	15	7	0.01 s	7	0.00 s
6	17	8	0.70 s	9	0.00 s
7	18	7	0.03 s	7	0.00 s
8	22	9	1.30 s	9	0.01 s
9	26	9	25.77 s	10	0.00 s
10	22	8	0.62 s	9	0.00 s
11	22	10	7.88 s	10	0.00 s
12	23	9	1.96 s	10	0.01 s
13	23	9	2.36 s	9	0.01 s

Tab. 4.5: Test results for clinical matrices

5. FURTHER RESULTS

In this chapter we discuss some more results concerning the use of multileaf collimators without interleaf collision constraint. In the first section we propose to change the direction of leaf movement between the different segments and present some numerical results for a heuristic approach to the TNMU–minimization in this setup. A possible further step in this direction is discussed in the second section: one could use two MLC's, perpendicular to each other, so that there is a leaf pair for each row and a leaf pair for each column. The numerical tests presented here suggest that this might yield considerable savings in terms of the TNMU.

5.1 Using the MLC in two directions

Here and in the next section we are concerned only with the minimization of the TNMU, neglecting the number of segments. So for simplicity we assume that every segment has coefficient 1. In this section we search for a realization of the given intensity matrix A in the following setup: the multileaf collimator has no interleaf collision constraint and can be rotated about 90°. So there are two different types of segments, called horizontal and vertical segments, according to the choice of the direction of leaf motion. In the first subsection we derive a backtracking algorithm for the TNMU–minimization problem. Due to the computational complexity this algorithm is applicable only for small problem sizes (10×10–matrices with entries between 0 and 7), but in any case we obtain a lower bound for the TNMU by interrupting the backtracking after some time. In the second subsection these lower bounds are compared with heuristic results.

5.1.1 A lower bound

Theorem 6. *Let c_1, c_2 be nonnegative integers and put $c = c_1 + c_2$. Then a segmentation of A with c_1 horizontal and c_2 vertical segments exists iff $B := cJ - A$ (where J is the all–one–matrix of size $m \times n$) can be written as a sum of four nonnegative integer matrices $P = (p_{i,j})$, $Q = (q_{i,j})$, $R = (r_{i,j})$, $S = (s_{i,j})$ with the following properties.*

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

2. $p_{i,j} \geq p_{i,j+1}$, $q_{i,j} \leq q_{i,j+1}$ *for* $i \in [m]$, $j \in [n-1]$

3. $r_{i,j} \geq r_{i+1,j}$, $s_{i,j} \leq s_{i+1,j}$ *for* $i \in [m-1]$, $j \in [n]$

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

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

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

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

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

Finally, for all $k \in [c]$, put $s_{i,j}^{(k)} = \sigma_{i,j}^{(k)} \tau_{i,j}^{(k)}$. To conclude the proof we have to check that the matrices $S^{(k)} = (s_{i,j}^{(k)})$ are segments and that their sum is A. Condition 1 in the theorem implies that, for each k and each (i, j), at most one of the numbers $\sigma_{i,j}^{(k)}$ and $\tau_{i,j}^{(k)}$ is 0, corresponding to the fact that a bixel can not be covered by both leaves of a leaf pair at the same time. Thus the number of indices k with $s_{i,j}^{(k)} = 0$ equals $p_{i,j} + q_{i,j} + r_{i,j} + s_{i,j} = b_{i,j}$, hence

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

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

Example 12. For $A = \begin{pmatrix} 1&4&2&5\\ 1&3&3&2\\ 1&3&2&5\\ 6&4&6&0 \end{pmatrix}$ and $c = 6 = 4 + 2$ we have

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

This yields

$$\sigma^{(1)} = \begin{pmatrix} 0&0&0&1\\ 0&0&1&1\\ 0&0&1&1\\ 1&1&1&1 \end{pmatrix}, \qquad \tau^{(1)} = \begin{pmatrix} 1&1&1&1\\ 1&1&1&1\\ 1&1&1&1\\ 1&1&1&0 \end{pmatrix}, \qquad S^{(1)} = \begin{pmatrix} 0&0&0&1\\ 0&0&1&1\\ 0&0&1&1\\ 1&1&1&0 \end{pmatrix},$$

$$\sigma^{(2)} = \begin{pmatrix} 0&0&0&1\\ 0&1&1&1\\ 0&0&1&1\\ 1&1&1&1 \end{pmatrix}, \qquad \tau^{(2)} = \begin{pmatrix} 1&1&1&1\\ 1&1&1&1\\ 1&1&1&1\\ 1&1&1&0 \end{pmatrix}, \qquad S^{(2)} = \begin{pmatrix} 0&0&0&1\\ 0&1&1&1\\ 0&0&1&1\\ 1&1&1&0 \end{pmatrix},$$

$$\sigma^{(3)} = \begin{pmatrix} 0&1&1&1\\ 1&1&1&1\\ 0&1&1&1\\ 1&1&1&1 \end{pmatrix}, \qquad \tau^{(3)} = \begin{pmatrix} 1&1&1&1\\ 1&1&1&0\\ 1&1&1&1\\ 1&1&1&0 \end{pmatrix}, \qquad S^{(3)} = \begin{pmatrix} 0&1&1&1\\ 1&1&1&0\\ 0&1&1&1\\ 1&1&1&0 \end{pmatrix},$$

$$\sigma^{(4)} = \begin{pmatrix} 1&1&1&1\\ 1&1&1&1\\ 1&1&1&1\\ 1&1&1&1 \end{pmatrix}, \qquad \tau^{(4)} = \begin{pmatrix} 1&1&1&1\\ 0&0&0&0\\ 1&1&1&1\\ 1&1&1&0 \end{pmatrix}, \qquad S^{(4)} = \begin{pmatrix} 1&1&1&1\\ 0&0&0&0\\ 1&1&1&1\\ 1&1&1&0 \end{pmatrix},$$

$$\sigma^{(5)} = \begin{pmatrix} 0&1&0&0\\ 0&1&0&0\\ 0&1&0&1\\ 1&1&1&1 \end{pmatrix}, \qquad \tau^{(5)} = \begin{pmatrix} 1&1&1&1\\ 1&1&1&1\\ 1&1&1&1\\ 1&0&1&0 \end{pmatrix}, \qquad S^{(5)} = \begin{pmatrix} 0&1&0&0\\ 0&1&0&0\\ 0&1&0&1\\ 1&0&1&0 \end{pmatrix},$$

$$\sigma^{(6)} = \begin{pmatrix} 0&1&0&1\\ 0&1&0&1\\ 0&1&0&1\\ 1&1&1&1 \end{pmatrix}, \qquad \tau^{(6)} = \begin{pmatrix} 1&1&1&1\\ 1&0&1&0\\ 1&0&1&0\\ 1&0&1&0 \end{pmatrix}, \qquad S^{(6)} = \begin{pmatrix} 0&1&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 1&0&1&0 \end{pmatrix}.$$

Based on Theorem 6, Algorithm 4 finds the minimal TNMU c and a decomposition of $B = cJ - A$ corresponding to a segmentation with this TNMU. The step in line 7 can be realized using backtracking. Of course this method is very time–consuming. Our implementation solved the problem for 10×10–matrices with random entries between 0 and 7 in a few seconds, but for larger problems this algorithm is not practicable. By interrupting the backtracking when some fixed time limit is reached without getting a construction or a contradiction in line 7, we still obtain a lower bound for the TNMU.

5.1.2 Heuristic results

For notational convenience we add a $0-$th and an $(m+1)-$th row with $a_{0,j} = a_{m+1,j} = 0$ $(j \in [n])$, and put

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

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

Algorithm 4 Minimal TNMU for 2–directional segmentation

$c = \max\{a_{i,j} \ : \ (i,j) \in [m] \times [n]\}$
finished:=false
while not finished **do**
  $B := cJ - A$
  $c_1 := c$; $c_2 := 0$
  while not finished **and** $c_1 \geq 0$ **do**
    Construct a decomposition of B as in Theorem 6 or derive a contradiction from the assumption that such a decomposition exists
    if Construction successful **then**
      finished:=true
    else
      $c_1 := c_1 - 1$; $c_2 := c_2 + 1$
    end if
  end while
  if not finished **then**
    $c := c + 1$
  end if
end while

Let also

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

According to [10] there is a segmentation without vertical segments with $c_{hor}(A)$ monitor units, and there is a segmentation without horizontal segments with $c_{vert}(A)$ monitor units. Hence an upper bound for the minimal number of monitor units needed to realize A is $c(A) = \min\{c_{hor}(A), c_{vert}(A)\}$.

As in Chapter 2 we construct a segmentation by successively subtracting segments until the zero matrix is reached. For the choice of the segment S we suggest a heuristic method. First $c_{hor}(A)$ and $c_{vert}(A)$ are determined and the direction of leaf motion is chosen to be horizontal if $c_{hor}(A) \leq c_{vert}(A)$ and vertical otherwise. We describe the construction of S for the horizontal case, the vertical case is treated analogously. For the new matrix $A' := A - S$, we have two aims. Firstly, we want that

$$c_{hor}(A') = c_{hor}(A) - 1, \tag{5.1}$$

and secondly $c_{vert}(A')$ should be small. Let us consider the second condition first. With the segments we associate (D, D')–paths in a layered digraph

$\Gamma = (V, E)$ similar to the digraph Γ in Chapter 3:

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

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

$$w(D, (1, l, r)) = \sum_{j=l}^{r} \max\{1, \beta(j) + 5 - c_{vert}(A)\} \qquad (1 \le l \le r + 1 \le n)$$
$$w((m, l, r), D') = 0 \qquad (1 \le l \le r + 1 \le n + 1)$$

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

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

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

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

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

Example 13. Let rows $i-1$ and i of matrix A be

$$\left(\begin{smallmatrix} 2 & 4 & 1 & \mathbf{4} & \mathbf{3} & \mathbf{4} & \mathbf{3} & 3 \\ 1 & \mathbf{5} & \mathbf{3} & \mathbf{2} & \mathbf{7} & \mathbf{4} & 3 & 2 \end{smallmatrix}\right),$$

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

$$\left(\begin{smallmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{smallmatrix}\right).$$

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

$$\left(\begin{smallmatrix} 2 & 4 & 1 & \mathbf{3} & \mathbf{2} & \mathbf{3} & \mathbf{2} & 3 \\ 1 & \mathbf{4} & \mathbf{2} & \mathbf{1} & \mathbf{6} & \mathbf{3} & 3 & 2 \end{smallmatrix}\right)$$

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

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

Then

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

and we obtain

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

Here the positive terms correspond to

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

while the negative term corresponds to

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

Finally, we delete all vertices (i, l, r) that lead to segments violating (5.1). Fix some vertex (i, l, r) and put

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

A segment corresponding to a path through (i, l, r) can not satisfy (5.1) if $\alpha'(i) := \sum_{j=1}^{n} \max\{0, a'_{i,j} - a'_{i,j-1}\} > c_{hor}(A) - 1$. But the only terms in $\alpha'(i)$ that could be different from the corresponding terms in $\alpha(i)$ are the terms for $j = l$ and $j = r + 1$ (if $r < n$), and for these terms we have, if $l \le r$,

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

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

So

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

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

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

Now we choose a segment S corresponding to a $(D, D')-$path of maximal weight. In Table 5.1 the results of the heuristic are compared to the average lower bound obtained by the backtracking method from Subsection 5.1.1 and to the optimal TNMU when the MLC is used in only one direction. On a 2 GHz workstation the computation times for the whole heuristic columns were 172 seconds (10×10) and 926 seconds (15×15). For the clinical matrices our

	10×10			15×15		
L	TNMU (old)	TNMU (new)	Lower bound	TNMU (old)	TNMU (new)	Lower bound
3	9.8	7.2	6.7	14.0	10.0	8.3
4	12.6	9.0	8.3	17.9	12.4	10.2
5	15.5	10.8	10.1	21.7	14.8	12.3
6	18.1	12.6	11.8	25.6	17.2	14.5
7	20.8	14.3	13.5	29.4	19.6	16.2
8	23.6	15.9	14.7	33.2	21.9	18.0
9	26.4	17.7	16.6	37.0	24.2	20.1
10	29.0	19.5	18.4	40.9	26.6	21.8
11	31.8	21.3	19.6	44.7	29.0	24.0
12	34.5	23.0	21.9	48.5	31.4	25.7
13	36.9	24.6	22.8	52.3	33.7	27.8
14	39.8	26.3	24.8	56.2	36.1	29.5
15	42.4	28.2	26.2	59.8	38.3	31.3
16	45.2	29.8	27.9	63.3	40.7	32.6

Tab. 5.1: Average TNMU for random 10×10– and 15×15–matrices with maximal entry L when the MLC is used in both directions (column 'TNMU(new)'), compared to the average lower bound and the results for the one–directional usage of the MLC (column 'TNMU(old)').

implementation of Algorithm 4 yields the optimal solution in a reasonable time and the results are shown in Table 5.2.

5.2 Two orthogonal MLCs

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

$$\begin{pmatrix} 0&1&1&0&1&0 \\ 1&0&1&1&1&1 \\ 1&0&1&1&0&0 \\ 1&0&1&1&1&1 \\ 1&0&1&0&0&0 \\ 1&0&0&0&1&1 \end{pmatrix} \tag{5.2}$$

no.	MU (new)	MU (old)	CPU–time
1	11	16	0.48 s
2	11	16	0.27 s
3	12	20	0.54 s
4	12	19	50.06 s
5	13	15	0.18 s
6	12	17	0.19 s
7	13	18	0.91 s
8	14	22	1.05 s
9	16	26	4.04 s
10	14	22	13.07 s
11	15	22	93.68 s
12	16	23	1.59 s
13	14	23	201.07 s

Tab. 5.2: Test results for the clinical matrices: the TNMU when the MLC is used in both directions ('MU (new)') compared with the TNMU when the MLC is used in only one direction ('MU (old)'), and the time needed to determine the segmentation according to Algorithm 4

.

is a segment corresponding to the leaf positions shown in Figure 5.1. The LP–formulation of the TNMU–segmentation problem given in Section 2.1 is still valid, we just have a larger set $\mathcal{F}$ of allowed segments, and correspondingly there are more inequalities that have to be satisfied by a dual feasible solution g. Here we suggest a heuristic segmentation method based on dual feasible solutions of a particular type. For pairs (i,j) with $2 \leq i \leq m-1$ and $2 \leq j \leq n-1$ we put

$$\gamma_1(i,j) = \max\left\{\frac{1}{3}(a_{i,s}+a_{i,t}+a_{u,j}+a_{v,j}-a_{i,j}) \ :\ \substack{1\leq s<j<t\leq n,\\ 1\leq u<i<v\leq m}\right\}. \qquad (5.3)$$

Then

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

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

$$g(i,s) = g(i,t) = g(u,j) = g(v,j) = \frac{1}{3}, \quad g(i,j) = -\frac{1}{3}$$

and $g(p,q) = 0$ for all other $(p,q) \in [m]\times[n]$. This is illustrated in Figure 5.2. If we choose s, t, u and v so that the maximum in (5.3) is attained the

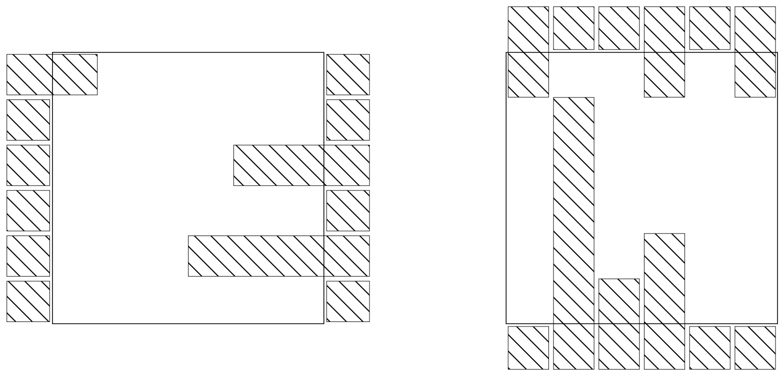


Fig. 5.1: Leaf positions corresponding to the segment (5.2).

$$\begin{array}{cccccc} & & & j & & \\ & & & \frac{1}{3} & & u \\ i & & \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} & \\ & & & \frac{1}{3} & & v \\ & & s & & t & \end{array}$$

Fig. 5.2: The structure of the dual feasible solution g, where the empty spaces are filled with zeros.

objective value of g is $\gamma_1(i,j)$ and to see that g is feasible we just have to observe that in order to cover the bixel (i,j) we have to cover at least one of the bixels (i,s), (i,t), (u,j), (v,j). Of course, $\tilde{c}(A)$ does not have to be an integer, and since we are using only integer coefficients even $\lceil \tilde{c}(A) \rceil$ can be used as a lower bound for the TNMU. As in Chapter 2, our algorithm is based on the general principle of extracting segments from A. Precisely, depending on A we determine a segment S, such that

$$A' = A - S$$

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

$$a_{i,j} = 0 \quad \Longrightarrow \quad s_{i,j} = 0 \qquad (i \in [m],\ j \in [n]) \tag{5.4}$$

is a necessary and sufficient condition for the nonnegativity of $A' = A - S$. Now we start with S equal to the all–one matrix of dimension $m \times n$ and cover successively all the zero–entries of A. The construction of S is described in Algorithm 5. Lines 2 to 9 ensure that condition (5.4) is satisfied for bixels

Algorithm 5 Segment for two orthogonal MLCs

$s_{i,j} := 1$ $(i \in [m],\ j \in [n])$
for $j = 1$ to n **do**
  if $a_{1,j} = 0$ **then** $s_{1,j} = 0$
  if $a_{m,j} = 0$ **then** $s_{m,j} = 0$
5: **end for**
for $i = 1$ to m **do**
  if $a_{i,1} = 0$ **then** $s_{i,1} = 0$
  if $a_{i,n} = 0$ **then** $s_{i,n} = 0$
end for
10: **while** (5.4) is violated **do**
  Choose an (i,j) with $a_{i,j} = 0$ and $s_{i,j} = 1$
  Choose a covering direction from $\{\text{left}, \text{right}, \text{up}, \text{down}\}$
  Cover bixel (i,j) with the leaf from the chosen direction
  Let S be the segment corresponding to the new leaf positions
15: **end while**

(i,j) with $i \in \{1,m\}$ or $j \in \{1,n\}$. These bixels can be covered without influencing other bixels, for instance $(i,1)$ can be covered by the left leaf of row i, so lines 2 to 9 imply no loss of generality. We still have to make precise, how we choose the bixel (i,j) in line 11 and the direction from which we cover it in line 12. For the choice of the bixel we follow the strategy to cover bixels (i,j) with a high value of $\gamma_1(i,j)$ first. Suppose we have already chosen the bixel to cover. Now we have to choose the direction from which we want to cover it. For instance by covering bixel (i,j) by the left leaf we also cover all bixels (i,j') with $1 \le j' \le j$. So it is natural to cover bixel (i,j) from a direction with the property that the maximal entry of a bixel that is covered although the entry is nonzero, is as small as possible. Let

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

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

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

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

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

and choose for $dir(i,j)$ the direction with in first instance minimal value of $\gamma_2^{(*)}(i,j)$, and in second instance maximal value of $\gamma_3^{(*)}(i,j)$. If there is still a tie it can be decided randomly. Finally, we put

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

We choose among the bixels (i,j) with $a_{i,j} = 0$ and $s_{i,j} = 1$ one with in first instance maximal value of $\gamma_1(i,j)$, and in second instance maximal value of

$\gamma_3(i,j)$. Now the chosen bixel (i,j) is covered by the leaf that is given by $dir(i,j)$. Precisely, we put

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

This covering step is repeated until (5.4) is satisfied.

Example 14. Consider the matrix

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

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

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

and continuing we obtain a segmentation with 5 segments.

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

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

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

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

Table 5.3 shows the average TNMU obtained by our algorithm, the average lower bound $\tilde{c}(A)$ and the optimal TNMU when only one MLC is used in one direction. On a 2 GHz workstation the computation for the whole 10×10–column took 91 seconds and for the 15×15–column it took 610 seconds. The results for the clinical sample matrices are shown in Table 5.4.

	10×10			15×15		
L	TNMU (old)	TNMU (new)	Lower bound	TNMU (old)	TNMU (new)	Lower bound
3	9.8	5.4	3.9	14.0	7.4	4.0
4	12.6	6.7	5.2	17.9	9.0	5.3
5	15.5	8.0	6.4	21.7	10.5	6.7
6	18.1	9.2	7.6	25.6	12.0	7.9
7	20.8	10.4	8.8	29.4	13.6	9.3
8	23.6	11.7	10.0	33.2	15.1	10.5
9	26.4	12.9	11.2	37.0	16.6	11.8
10	29.0	14.1	12.4	40.9	18.1	13.0
11	31.8	15.4	13.6	44.7	19.6	14.2
12	34.5	16.6	14.8	48.5	21.1	15.7
13	36.9	17.9	16.0	52.3	22.5	16.9
14	39.8	19.1	17.1	56.2	24.1	18.1
15	42.4	20.3	18.3	59.8	25.5	19.6
16	45.2	21.5	19.5	63.3	27.0	20.4

Tab. 5.3: Average TNMU for random 10×10– and 15×15–matrices mith maximal entry L when two orthogonal MLCs are used (column 'TNMU(new)'), compared to the average lower bound and the results when one MLC is used in only one direction (column 'TNMU(old)').

no.	MU (1)	MU (2)	MU (3)	Bound	CPU–time
1	16	11	11	11	0.00 s
2	16	11	11	11	0.00 s
3	20	12	11	11	0.00 s
4	19	12	11	10	0.00 s
5	15	13	10	10	0.00 s
6	17	12	12	11	0.01 s
7	18	13	11	11	0.00 s
8	22	14	13	12	0.01 s
9	26	16	13	12	0.01 s
10	22	14	14	13	0.01 s
11	22	15	14	13	0.01 s
12	23	16	14	12	0.02 s
13	23	14	14	12	0.02 s

Tab. 5.4: Test results for clinical matrices: the TNMU with one MLC in one direction ('MU (1)'), with one MLC in two directions ('MU (2)') and with two orthogonal MLCs ('MU (3)'), the lower bounds for the last case ('Bound') and the times needed to determine the segmentations for two MLCs with our algorithm.

BIBLIOGRAPHY

[1] M. Aigner. *Diskrete Mathematik.* Vieweg, Braunschweig/Wiesbaden, 4th edition, 2001.

[2] D. Baatar and H.W. Hamacher. New LP model for multileaf collimators in radiation therapy. contribution to the conference ORP3, University of Kaiserslautern, 2003.

[3] T. Benoist and F. Chauvet. Complexity of some FPP related problems. Technical report, Bouygues' e–lab, 2001.

[4] 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.

[5] I.M. Bomze and W. Grossmann. *Optimierung – Theorie und Algorithmen.* BI–Wissenschaftsverlag, Mannheim, 1993.

[6] 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.

[7] A.L. Boyer and C.Y. Yu. Intensity–modulated radiation therapy with dynamic multileaf collimators. *Semin. Radiat. Oncol.*, 9:48–59, 1999.

[8] J. Dai and Y. Zhu. Minimizing the number of segments in a delivery sequence for intensity–modulated radiation therapy with a multileaf collimator. *Med. Phys.*, 28:2113–2120, 2001.

[9] M.L.P. Dirkx, B.J.M. Heijmen, and J.P.C. van Santvoort. Leaf trajectory calculation for dynamic multileaf collimation to realize optimized fluence profiles. *Phys. Med. Biol.*, 43(8):1171–1184, 1998.

[10] K. Engel. A new algorithm for optimal multileaf collimator field segmentation. Preprint 03/5, Fachbereich Mathematik, Uni Rostock, under revision for *Discr. Appl. Math.*, 2003.

[11] 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.

[12] M.R. Garey and D.S. Johnson. *Computers and intractability, a guide to the theory of NP–completeness*. W.H. Freeman, 1979.

[13] A. Holder and B. Salter. A tutorial on radiation oncology and optimization. Technical Report 86, Trinity University, San Antonio, Texas, 2004.

[14] D. Jungnickel. *Graphen, Netzwerke und Algorithmen*. BI–Wissenschaftsverlag, Mannheim, 1994.

[15] T. Kalinowski. Realization of intensity modulated radiation fields using multileaf collimators. In R. Ahlswede, with the assistance of L. Bäumer, and N. Cai, editors, *General Theory of Information Transfer and Combinatorics*. Shannon Foundation, to be published 2004. Report on a Research Project at the ZIF (Center of interdisciplinary research) in Bielefeld Oct. 1, 2002 – August 31, 2003.

[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, and S. Ranka. Algorithms for optimal sequencing of dynamic multileaf collimators. *Phys. Med. Biol.*, 49(1):33–54, 2004.

[18] M. Langer, V. Thai, and L. Papiez. Improved leaf sequencing reduces segments of monitor units needed to deliver IMRT using multileaf collimators. *Med. Phys.*, 28:2450–2458, 2001.

[19] L. Ma, A.L. Boyer, L. Xing, and C.M. Ma. An optimized leaf setting algorithm for beam intensity modulation using dynamic multileaf collimators. *Phys. Med. Biol.*, 43(6):1629–1643, 1998.

[20] W. Que. Comparison of algorithms for multileaf collimator field segmentation. *Med. Phys.*, 26:2390–2396, 1999.

[21] I.I. Rosen, R.G. Lane, S.M. Morill, and J.A. Belli. Treatment plan optimization using linear programming. *Med. Phys.*, 18:141–152, 1991.

[22] D. Shepard, M. Ferris, G. Olivera, and T. Mackie. Optimizing the delivery of radiation therapy to cancer patients. *SIAM Review*, 41(4):721–744, 1999.

[23] 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.

[24] S.V. Spirou and C.S. Chui. Generation of arbitrary intensity profiles by dynamic jaws or multileaf collimators. *Med. Phys.*, 21:1031–1041, 1994.

[25] R. Svensson, P. Källman, and A. Brahme. An analytical solution for the dynamic control of multileaf collimators. *Phys. Med. Biol.*, 39(1):37–61, 1994.

[26] J.P.C. van Santvoort and B.J.M. Heijmen. Dynamic multileaf collimation without 'tongue-and-groove' underdosage effects. *Phys. Med. Biol.*, 41(10):2091–2105, 1996.

[27] S. Webb, T. Bortfeld, J. Stein, and D. Convery. The effect of stair–step leaf transmission on the 'tongue–and–groove problem' in dynamic radiotherapy with a multileaf collimator. *Phys. Med. Biol.*, 42(3):595–602, 1997.

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

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfaßt, andere als die von mir angegebenen Quellen nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Thomas Kalinowski

LEBENSLAUF

Thomas Kalinowski
geb. am 27.11.1980 in Pasewalk
Familienstand: ledig

Schulbildung

09/86 – 08/92	:	Polytechnische Oberschule in Grambow
09/92 – 08/94	:	Gymnasium Löcknitz
09/94 – 07/98	:	Abitur an der CJD–Jugenddorf– Christophorusschule Rostock

Studien und akademische Ausbildung

10/99 – 09/03	:	Mathematikstudium an der Universität Rostock (Nebenfach Informatik), Stipendiat der Studienstiftung des deutschen Volkes
09/01 – 04/02	:	Studienaufenthalt als ERASMUS–Stipendiat an der University of Glasgow (Schottland)
09/03	:	Abschluß des Diplomstudiums, Thema der Diplomarbeit: Realization of intensity modulated radiation fields using multileaf collimators

Berufliche Tätigkeit

Seit 10/03	:	Wissenschaftlicher Mitarbeiter am Fachbereich Mathematik der Universität Rostock

Schülerförderung

Seit 09/00	:	Mitglied im Mathematik–Olympiaden e.V.
09/00 – 07/01	:	Leitung einer Mathematik–AG für Schüler der Klassen 9–10
09/02 – 07/03	:	Leitung einer Mathematik–AG für Schüler der Klassen 11–12

Thesen zur Dissertation

von Thomas Kalinowski

1. In der Krebsbehandlung werden Mehrlamellenkollimatoren eingesetzt um intensitätsmodulierte Bestrahlungsfelder zu erzeugen. Ein Schritt bei der Erstellung eines Behandlungsplans ist die Realisierung einer gegebenen Fluenzverteilung mit einer geringen Gesamtbestrahlungsdauer und einer kleinen Anzahl von homogenen Feldern. Die möglichen Lamellenpositionen des Kollimators können durch $(0,1)-$Matrizen, sogenannte Segmente, beschrieben werden. Damit lässt sich das Problem als Suche nach einer Segmentierung einer nichtnegativen ganzzahligen $m \times n$–Matrix A formulieren, d.h. nach einer Darstellung

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

von A als positive Linearkombination von Segmenten $S_1, \dots, S_k$.

2. In vielen Mehrlamellenkollimatoren verbietet die Lamellenkollisionsbedingung (interleaf collision constraint) das Überlappen von gegenüberliegenden Lamellen in benachbarten Zeilen. Für diese Kollimatoren sind die $m \times n$–Segmente $S = (s_{i,j})$ charakterisiert durch die Existenz von ganzen Zahlen l_i, r_i $(i = 1, \dots, m)$, so dass

$$\begin{aligned}
& l_i \leq r_i + 1 && (i \in [m]), \\
& s_{i,j} = \begin{cases} 1 & \text{falls } l_i \leq j \leq r_i \\ 0 & \text{sonst} \end{cases} && (i \in [m],\ j \in [n]), \\
& l_i \leq r_{i+1} + 1,\ r_i \geq l_{i+1} - 1 && (i \in [m-1]).
\end{aligned}$$

3. Die Minimierung der Gesamtbestrahlungsdauer wird als lineares Optimierungsproblem formuliert:

$$(P) \left\{ \begin{array}{rll}
\displaystyle\sum_{S \in \mathcal{S}} f(S) & \rightarrow \ \min & \\
f(S) & \geq 0 & \forall S \in \mathcal{S}, \\
\displaystyle\sum_{S \in \mathcal{S} : (i,j) \in S} f(S) & = a_{i,j} & \forall (i,j) \in V.
\end{array} \right.$$

Hier ist $\mathcal{S}$ die Menge der Segmente, wobei ein Segment S mit der Menge der Paare $(i,j) \in [m] \times [n]$ mit $s_{i,j} = 1$ identifiziert wird.

4. Mit Hilfe des dualen Problems

$$(D)\left\{\begin{array}{rll} \sum\limits_{(i,j)\in[m]\times[n]} a_{i,j}g(i,j) & \rightarrow \max & \\ & & \\ \sum\limits_{(i,j)\in S} g(i,j) & \leq 1 & \forall S \in \mathcal{S} \end{array}\right.$$

wird der minimale Wert der Zielfunktion im Problem (P) als maximales Gewicht eines $0-1-$Weges in folgendem gewichteten Digraphen $G = (V, E)$ charakterisiert:

$$\begin{aligned} V &= [m] \times [n] \cup \{0,1\}, \\ E &= E_1 \cup E_2 \quad \text{mit} \\ E_1 &= \{(0,(i,1)) \ : \ i \in [m]\} \cup \{((i,n),1) \ : \ i \in [m]\}, \\ E_2 &= \{((i,j),(i',j+1)) \ : \ i,i' \in [m], j \in [n-1]\}, \end{aligned}$$

mit der Gewichtsfunktion

$$\begin{aligned} \delta(0,(i,1)) &= a_{i,1} & & (i \in [m]), \\ \delta((i,n),1) &= 0 & & (i \in [m]), \\ \delta((i,j),(i,j+1)) &= \max\{0, a_{i,j+1} - a_{i,j}\} & & (i \in [m], j \in [n-1]), \\ \delta((i,j),(i',j+1)) &= \max\{0, a_{i',j+1} - a_{i',j}\} - \sum_{k=i}^{i'-1} a_{k,j} & & \\ & & & (i,i' \in [m], i < i', j \in [n-1]), \\ \delta((i,j),(i',j+1)) &= \max\{0, a_{i',j+1} - a_{i',j}\} - \sum_{k=i'+1}^{i} a_{k,j} & & \\ & & & (i,i' \in [m], i > i', j \in [n-1]). \end{aligned}$$

5. Aus dem Beweis der Min–Max–Aussage in These 4 wird ein effizienter Algorithmus zur Bestimmung einer optimalen Lösung des Problems (P) abgeleitet.

6. Sei $c(A)$ die minimale Koeffizientensumme einer Segmentierung von A. Ein Paar (u, S) aus einer natürlichen Zahl u und einem Segment S heiße zulässig, falls $A' := A - uS$ nichtnegativ ist und $c(A') = c(A) - u$ gilt. Zur Konstruktion einer Segmentierung mit minimaler Koeffizientensumme und kleiner Segmentanzahl wird die folgende Greedy–Strategie verwendet.

1. Bestimme ein zulässiges Paar (u, S) mit maximalem u.

2. Setze $A := A - uS$.
3. Falls $A \neq 0$, gehe zu Schritt 1.

7. Die Minimierung der Segmentanzahl ist bereits für $m = 1$ streng NP–schwer. Es wird jedoch bewiesen, dass dieses Problem in Linearzeit gelöst werden kann, wenn die Matrixeinträge durch eine Konstante beschränkt sind.

8. Für $m > 1$ wird ein Algorithmus der Komplexität $O(mn^{2L+2})$ angegeben, der für einen Kollimator ohne Lamellenkollisionsbedingung eine Segmentierung mit minimaler Gesamtbestrahlungsdauer und minimaler Segmentanzahl bestimmt. Hier ist $L = \max\{a_{i,j} \ : \ (i,j) \in [m]\times[n]\}$.

9. Eine deutliche Reduzierung der Gesamtbestrahlungsdauer ist möglich, wenn der Kollimator zwischen der Verabreichung der einzelnen Felder um 90° gedreht werden darf. Hierzu wird eine Heuristik implementiert und getestet.

10. Eine weitere Reduzierung der notwendigen Gesamtbestrahlungsdauer kann durch die Verwendung von zwei zueinander orthogonal angeordneten Kollimatoren erreicht werden, wie wiederum mittels Implementierung eines heuristischen Algorithmus gezeigt wird.