Universität Rostock Traditio et Innovatio

Approximated multileaf collimator field segmentation

Dissertation

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

vorgelegt von
Antje Kiesel, geb. am 17.12.1983 in Bützow
aus Rostock

Rostock, 28. September 2010

urn:nbn:de:gbv:28-diss2011-0036-1

Dekan: Prof. Dr. Christoph Schick

Gutachter: Prof. Dr. K. Engel (Universität Rostock)

Gutachter: Prof. Dr. Horst W. Hamacher (Universität Kaiserslautern)

Gutachter: Doz. Dr. Samuel Fiorini (Université Libre de Bruxelles)

Tag der öffentlichen Verteidigung: 20. Januar 2011

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Prof. Dr. Konrad Engel for the very valuable support not only during my years as a diploma and doctoral student. You started to encourage me already in my school time and I benefited from your outstanding efforts during my whole studies. This support made this thesis, conference attendances and a number of publications possible. It was always a pleasure to work under your supervision.

I thank my colleague Dr. Thomas Kalinowski for many helpful discussions, for listening to my ideas and questions as well as for proofreading my papers. I really enjoyed the joint work with you and was always impressed by your great mathematical expertise.

I would especially like to thank Dr. Tobias Gauer, who parallel worked on his PhD thesis at the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf. Thank you for spending so much time with the computations for my projects and for giving me the knowledge about the physical background of my thesis. I learned a lot from you.

I also thank Prof. Dr. Samuel Fiorini and Dr. Céline Engelbeen for a wonderful collaboration and a very fruitful research visit that I made at the Université Libre de Bruxelles in March 2009. This supported the progress of my work very much and our results ended up in two joint publications. It was great to work again with you in November 2009 in Rostock. And thank you Céline for our wonderful friendship. Further thanks go to Prof. Dr. Horst W. Hamacher for comments and helpful suggestions in the course of preparing this thesis.

Warmest thanks to my parents, family and, especially, to my husband Thomas, for their overwhelming support and patience during the last years. Thank you for caring for our little son Jonas during the last weeks when I completed this thesis. And thank you Jonas, you make me so proud and happy!

Last but not least, I had the great honor to become a scholarship holder of the Studienstiftung des deutschen Volkes and I am grateful for the financial support and all the encouragement beyond study.

KURZFASSUNG

In der intensitätsmodulierten Strahlentherapie (IMRT) werden Multileaf-Kollimatoren zur Feldformung eingesetzt. Durch Überlagerung unterschiedlich geformter Felder versucht man, vorgeschriebene Dosisverteilungen möglichst gut zu realisieren. Diese Aufgabe kann als diskretes Optimierungsproblem modelliert werden, wobei gegebene nichtnegative ganzzahlige Matrizen durch eine nichtnegative ganzzahlige Linearkombination von gewissen 0-1-Matrizen (Segmenten) dargestellt werden müssen. Je nachdem welche technischen und dosimetrischen Nebenbedingungen in das Modell einfließen sollen, ergeben sich Probleme zur exakten beziehungsweise approximativen Zerlegung. Zudem können verschiedene Zielfunktionen der Optimierung, z.B. die Bestrahlungszeit oder der Approximationsfehler, betrachtet werden. Diese Probleme werden in der vorliegenden Arbeit vorgestellt und mit Methoden der kombinatorischen und ganzzahligen Optimierung sowie der Graphentheorie behandelt.

Zusätzlich wird im letzten Teil der Arbeit ein kontinuierliches Fluenzmodell eingeführt, bei dem die Zielfluenzen reelle nichtnegative Funktionen von zwei Variablen sind. Sinn dieses Ansatzes ist die bessere Modellierung der Eigenschaften von Strahlung, insbesondere die Berücksichtigung von Halbschatteneffekten am Rand der Felder. Daraus wird ein quadratisches Optimierungsproblem abgeleitet.

Während Teile der Dissertation nur von mathematischem Interesse sind, befassen sich andere Teile explizit mit klinisch anwendbaren Algorithmen. Diese werden am Fallbeispiel getestet und die numerischen Auswertungen ausführlich dargestellt.

Diese Arbeit liefert eine umfassende Sammlung von Segmentierungsalgorithmen für die Vielzahl der in der IMRT auftretenden Probleme mit Nebenbedingungen sowie Komplexitätsanalysen für ausgewählte Problemklassen.

ABSTRACT

In intensity-modulated radiation therapy (IMRT) multileaf collimators are used for field shaping. One tries to realize prescribed fluence distributions as good as possible by a superposition of several differently shaped fields. This task can be modeled as a discrete optimization problem, where given nonnegative integer matrices have to be decomposed into a nonnegative integer linear combination of certain 0-1-matrices (segments). Depending on the technical and dosimetric constraints one wants to consider, exact or approximate decomposition problems arise. Furthermore, different objective functions of the optimization, e.g. the delivery time or the approximation error, are of interest. These problems are introduced in this thesis and solved using methods from combinatorial and discrete optimization as well as graph theory.

Furthermore, in the last part of the thesis, a continuous fluence model is introduced, where the target fluences are real-valued nonnegative functions of two variables. The aim of this approach is the improved modeling of the characteristics of radiation, especially the penumbra effects at the border of the radiation fields. A quadratic optimization problem is deduced.

Whereas parts of the thesis are just of mathematical interest, other parts explicitly deal with clinically applicable algorithms. These are tested for a clinical case and the numerical results are displayed.

This thesis provides a comprehensive collection of segmentation algorithms for the variety of constrained problems arising in IMRT as well as a complexity analysis for some of the problem classes.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

"Mathematics is as much an aspect of culture as it is a collection of algorithms."
Carl Boyer (1906-1976)

"But there is another reason for the high repute of mathematics: it is mathematics that offers the exact natural sciences a certain measure of security which, without mathematics, they could not attain."
Albert Einstein (1879-1955)

Radiation therapy planning for cancer patients is a complex task, where physicians, physicists and mathematicians have to cooperate in order to achieve the best treatment plan for each specific case. The aim is to completely destroy the tumorous cells (target volume) while preserving the surrounding healthy organs (organs at risk) from the radiation. The physician has to contour the tumor regions carefully and to prescribe dose constraints for the targets. The mathematicians have to provide the necessary algorithms for a variety of steps of the planning process such that the physicist is capable of developing a treatment planning system and calculate individual therapy plans for the patients. One of the widely accepted and applied methods for the treatment is intensity-modulated radiation therapy (IMRT), where radiation is delivered by a linear accelerator with a rectangular beam head and differently shaped fields are generated by the use of a multileaf collimator (MLC). Their superposition yields a fluence distribution that should be as close to the target fluence as possible. An MLC consists of a number of metal leaf pairs that can be shifted towards each other from left and right independently, such that parts of the rectangular field are covered from radiation, while the open region receives fluence. The whole treatment process consists of mainly three steps: As radiation can be delivered from different beam angles and other degrees of freedom like couch angles and energy of the radiation can be used, the first step is to find a set of main fields with fixed couch and gantry angle and to compute an intensity modulation for them, i.e. determine a target fluence that should be delivered in this setting. The second step is the segmentation step, where each target fluence has to be decomposed into a number of MLC shapes whose superposition approximates the target fluence as good as possible. In the third step, the members of the segmentation are used as candidates for the final treatment plan. A precise dose calculation is computed for each of them and the final choice of segments and their irradiation times is carried out in a further optimization step (cf. [25] for details of the planning process and [23] for a survey on optimization in IMRT).
Thomas Bortfeld, one of the pioneers of IMRT, once stated that the algorithmic side of "leaf sequencing has now become a mathematical playground. Whole sessions at

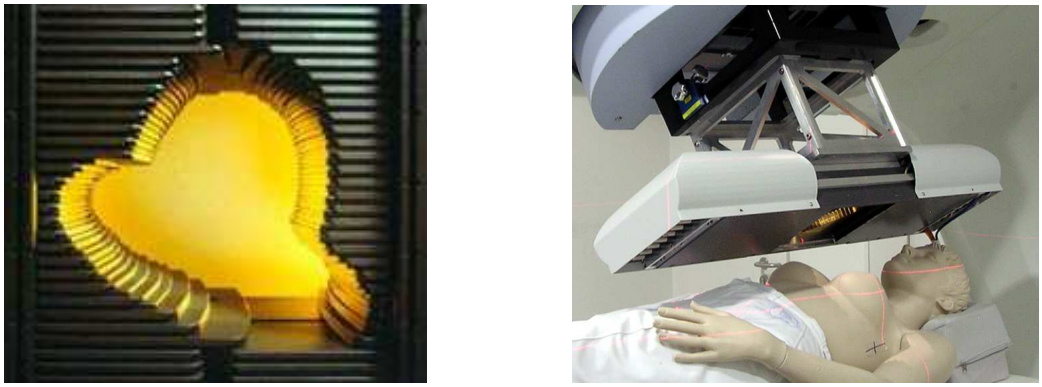

Fig. 1.1: Multileaf collimator and linear accelerator. Left: The leaf pairs of a multileaf collimator that can be shifted to form differently field shapes. Right: The linear accelerator and the treatment couch with gantry and beam head.

mathematical conferences have been devoted to the problem. However, the potential to make clinically relevant improvements in this particular field is rather limited." [10] Indeed, the number of abstract discrete and combinatorial optimization problems that were derived from the segmentation step is very large. A huge variety of side constraints on the field shapes and of objective functions is discussed. Some of them play an important role in clinical practice and have found application in modern planning systems. Others matter to mathematicians because the problems are interesting from a theoretical point of view and because they can be embedded into well-known mathematical contexts like graph theory or flow problems. And as the research in the leaf sequencing context has these two sides, this thesis has them as well. We deal with a number of optimization problems, divided into the three main classes of exact segmentation as well as discrete and continuous approximate segmentation. We are totally aware that parts are not clinically relevant, but from high theoretical interest. However, some parts of the thesis aim at modeling a realistic setup for segmentation and algorithms applicable for clinical practice are developed. This was possible due to a fruitful cooperation with the physicist Tobias Gauer from the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf, who worked parallel on his PhD thesis and developed a treatment planning system for electron beam therapy (cf. [35, 36, 37]). Some of the algorithms developed in this thesis were implemented into the treatment planning algorithm from [25], other approaches for the overall treatment planning process can for example be found in [54, 64, 65, 70].

Throughout the thesis, we restrict ourselves to the case where the MLC is used in the so called step-and-shoot mode, i.e. the radiation is switched off while the leaves are moving. There are other attempts using dynamic delivery of radiation [12, 19, 21, 46, 48, 56, 68]. Besides our sequential planning approach, so called aperture-based approaches [8, 25, 63, 66] aim at combining the intensity modulation and the segmentation step.

The next section embeds the segmentation task into the overall treatment planning process and introduces the different optimization problems that are considered in this thesis. The structure of the thesis is explained there in detail. We will make use of

linear and quadratic programming, graph theory, network flow problems, the theory of linear and nonlinear optimization, complexity theory and develop, if possible, optimal algorithms and, if not possible, heuristic algorithms for different (mainly approximate) segmentation problems.
The aim of the thesis is on the one hand to contribute new approaches to those exact decomposition problems, that are not solved satisfactory until now. On the other hand, there are various reasons that motivate the approximate segmentation tasks (that will be explained in detail later). Approximate segmentation problems have been considered only rarely yet and thus, the approaches in this thesis build a first comprehensive collection of algorithms related to this topic. Finally, especially the continuous segmentation problem should provide a clinically applicable model for segmentation.
Main parts of this thesis are based upon work published or submitted for publication and partly include collaborative work: Section 4.3 is joint work with Céline Engelbeen [30]. Section 5.2 is based on [45] and was developed in collaboration with Thomas Kalinowski. Joint work with Samuel Fiorini and Céline Engelbeen led to the results in Section 6.1 that were published in [29]. Section 6.4 is based on [52] and the results from Section 7 can be found in [51].

2. SEGMENTATION AS PART OF THE TREATMENT PLANNING PROCESS

An MLC field is a quadruple $F = (\varphi, \theta, \eta, P)$, where φ is the gantry angle, θ is the couch angle, η is the energy of the radiation and P is the sequence of leaf positions of the MLC in form of a pair of vectors for the left and right leaves. A weighted field is a pair (F, x_F) that associates a nonnegative delivery time of radiation x_F with the field F. The aim of the treatment planning process is to determine a sequence of weighted MLC fields whose superposition yields a fluence distribution that realizes as good as possible the prescribed doses in the body of the patient. A treatment plan is a set of weighted fields

$$(\mathcal{F}, \boldsymbol{x}) = \{(F, x_F) \mid F \in \mathcal{F}\}.$$

To evaluate a plan, the (abstract) patient is discretized into a finite set of voxels V. We restrict ourselves to voxels that belong to the planning target volumes (PTVs) and to the organs at risk (OARs). Assume we have lower and upper dose requirements for the voxels of the PTVs as well as upper dose constraints for the voxels of the OARs. Let for each voxel v the amount of dose that is delivered to voxel v by the field F be denoted by $D_F(v)$. We generally use the additivity assumption, that the total dose at voxel v in a treatment plan $(\mathcal{F}, \boldsymbol{x})$ is

$$D_{(\mathcal{F},\boldsymbol{x})}(v) = \sum_{F \in \mathcal{F}} x_F D_F(v).$$

It is possible to evaluate the plan by a quadratic objective function that penalizes the deviations between realized and target dose for each voxel. For fixed $\mathcal{F}$, a quadratic programming problem can be formulated to find the weights $\boldsymbol{x}$ that minimize the value of the objective function. These approaches are developed in [25], where a detailed description of the whole planning process is given. After deciding for a set of so called main fields with fixed couch and gantry angle, an intensity modulation is carried out for each of them. This is done by discretizing the irradiation field into $m \times n$ bixels and by solving the quadratic programming problem for fields where only a single bixel (i, j) of the irradiation field is open. Thus, one gets a coefficient a_{ij} for each bixel and finally an intensity matrix $A = (a_{ij})$ that has to be decomposed into MLC leaf positions. This so called segmentation step is the subject of this thesis. In all our discrete optimization problems, the setting is the following: We want to realize a target fluence given by an $m \times n$ intensity matrix A with nonnegative integer entries by superimposing segments that are deliverable by the MLC. Let, for integral m, ℓ and r, $[m] := \{1, 2, \ldots, m\}$ and $[\ell, r] := \{\ell, \ell + 1, \ldots, r - 1, r\}$. Furthermore, $x_+ := \max(0, x)$ for $x \in \mathbb{R}$. We will throughout use capital letters for matrices, small letters for the entries and bold small

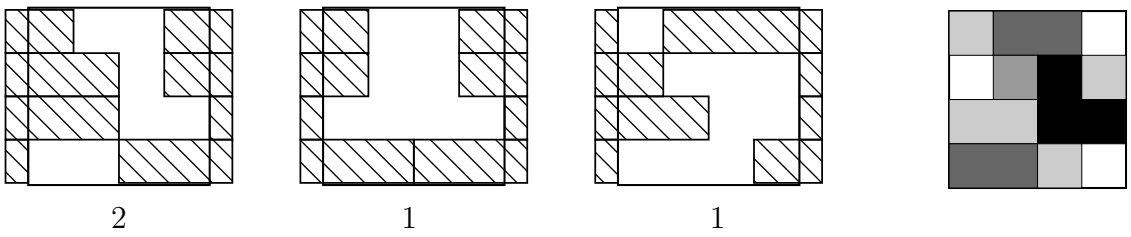

Fig. 2.1: Example of an MLC-segmentation. Leaf positions and irradiation times determining an exact decomposition of a 4×4 intensity matrix.

letters for vectors, e.g. a matrix A has the entries a_{ij} and the i-th row is denoted by $\boldsymbol{a}_i$. The segments are represented by 0-1-matrices of size $m \times n$ satisfying the *consecutive ones property*. A vector $\mathbf{v} \in \{0,1\}^d$ has the *consecutive ones property*, if $v_\ell = 1$ and $v_r = 1$ for $1 \leq \ell \leq r \leq d$ imply $v_j = 1$ for all $\ell \leq j \leq r$. A binary matrix S has the *consecutive ones property*, if each row of S has the consecutive ones property. That means, $S = (s_{ij})$ is a segment, if there are integral intervals $[\ell_i, r_i]$ for all $i \in [m]$ such that

$$s_{ij} = \begin{cases} 1, & \ell_i \leq j \leq r_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

The ones represent the uncovered region of the MLC whereas the zeros indicate the region that is covered by the leaves. For closed rows we use $\ell = r + 1$. Throughout the thesis, the positions (i, j) of a segment are called *bixels*. Let $\mathcal{S}$ be the set of all deliverable segments. An exact segmentation of the intensity matrix A is a decomposition

$$A = \sum_{S \in \mathcal{S}} u_S S,$$

where the u_S are nonnegative integer coefficients. Figure 2.1 illustrates how an MLC can be used to modulate the intensity. There have been various investigations into optimization problems concerning exact decompositions. The starting point was the unconstrained segmentation with the aim at minimizing the *delivery time* (DT) of a segmentation which is defined by

$$DT := \sum_{S \in \mathcal{S}} u_S$$

and measures the irradiation time for the patient. The unit of the delivery time is called a *monitor unit*, i.e. if the plan has a delivery time of 50, we say that 50 monitor units are needed. This problem is well studied and solutions can be found in [1, 6, 11, 40, 47, 60, 71]. It is desirable to minimize the delivery time, for instance, in order to reduce the side effects caused by diffusion of the radiation as much as possible and thus to avoid overdosage in the healthy tissues. Other attempts aim at minimizing the number of used the segments, i.e. $|\{S \in \mathcal{S} \mid u_S > 0\}|$ [4, 20, 24, 31, 43]. This problem as well as the lex-min-problem of finding a decomposition with minimal DT and minimal number of segments under this circumstance are NP-hard ([6, 41]) and

a heuristics is proposed in [5]. Minimizing the number of segments corresponds to keeping the total treatment time small which enables a high workload of the devices and a high throughput of patients in the hospitals. In practical applications, a number of constraints, that reduce the number of feasible segments, can be considered, i.e. one is interested in decompositions using only segments from a given subset $\mathcal{S}' \subseteq \mathcal{S}$. This subset might be given implicitly by a constraint (i.e. $\mathcal{S}'$ is the set of segments satisfying a given constraint) or explicitly by an enumeration of the allowed shapes. In this thesis, the interleaf collision constraint (ICC) [6, 9, 28, 39, 41, 45], the tongue-and-groove constraint (TG) [30, 42, 49, 50, 61], the minimum separation constraint (MSC) [29, 47] and the minimum field size constraint (MFC) [52, 53] will be discussed. There are also geometric approaches aiming at minimizing both the tongue-and-groove error and the number of used segments [17, 18]. In [28] the authors deal with another constraint called interleaf distance constraint. When ICC, TG or interleaf distance constraint are considered, exact decompositions are possible and the objective function of our optimization problem is the DT. For the remaining constraints, matrices are in general not decomposable and approximation problems arise. Section 6.1 is devoted to decompositions using an arbitrarily given set of segments $\mathcal{S}'$. We basically deal with three different types of problems for discrete segmentations here. Throughout, let $\mathcal{S}'$ be the set of allowed segments.

For **exact decompositions**, we consider problems of the form

MIN-DT: Find a decomposition $A = \sum_{S \in \mathcal{S}'} u_S S$ such that the DT $\sum_{S \in \mathcal{S}'} u_S$ is minimum.

For the **approximation problems**, we want to find approximate decompositions that are close to the target fluence. Let therefore $\|\cdot\|$ be a specified vector norm on $\mathbb{R}^k$. In this thesis, we use either the ℓ_1-norm or the ℓ_2-norm

$$\|\boldsymbol{v}\|_1 := \sum_{i=1}^{k} |v_i|, \quad \|\boldsymbol{v}\|_2 := \sqrt{\sum_{i=1}^{k} v_i^2}$$

to measure deviations between target and approximation vectors. Analogously, if we deal with matrices, we consider a matrix $A \in \mathbb{R}^{m \times n}$ as vector of size mn and define its norm as the norm of the corresponding vector:

$$\|A\|_1 := \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|, \quad \|A\|_2 := \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}.$$

Throughout the thesis, if nothing else is stated, the problems with ℓ_1-norm are considered. If the ℓ_2-norm is taken into account we will explicitly mention it. From a practical point of view, it is still unclear which norm should be preferred.
Basically, if the set $\mathcal{S}'$ does not enable exact decompositions of all matrices, the approximation problems we want to consider are of the form:

Approx-MIN-TC: Find a nonnegative integral approximation matrix B that is decomposable into segments from $\mathcal{S}'$ such that $\|A - B\|$ is minimum.

The value of the objective function is called *total change* (TC). Minimizing the total change corresponds to avoiding large under- or overdosage effects caused by the approximation. Section 6 basically deals with problems of type **Approx-MIN-TC** for various choices of $\mathcal{S}'$. We also give a column generation approach to the problem in Section 6.3 and provide a clinically relevant segmentation model in Section 6.4 with a slightly modified objective.
There is another type of approximation problems that arise in clinical applications. They can be considered in the unconstrained ($\mathcal{S}' = \mathcal{S}$) or constrained ($\mathcal{S}' \subset \mathcal{S}$) case. Sometimes, the DT of exact decompositions is too large to be acceptable and this leads to problems of the form:

Approx-MIN-DT: Given lower and upper bound nonnegative integral matrices $\underline{A}$ and $\overline{A}$, find a nonnegative integral approximation matrix B with $\underline{a}_{ij} \leq b_{ij} \leq \overline{a}_{ij}$ for all $(i,j) \in [m] \times [n]$ such that the DT of a segmentation of B into segments from $\mathcal{S}'$ is minimum.

Approx-MIN-DT-TC: Solve **Approx-MIN-DT** such that DT and TC are minimized lexicographically.

If the segments from $\mathcal{S}'$ do not enable an exact decomposition of a feasible matrix, the optimal delivery time is $DT = \infty$. Section 5 analyzes the unconstrained case and the case when the interleaf collision constraint is regarded. The justification to consider these problems is also that from a practical point of view there seem to be some doubts if it is reasonable to consider every entry a_{ij} as fixed once and for all. First, the matrix A is a result of numerical computations which are based on simplified physical models of how the radiation passes through the patients body, and second, the representation of A as a superposition of homogeneous fields is also based on model assumptions which are not strictly correct, for instance the dose delivered to an exposed bixel depends on the shape of the field. So it might be sufficient, to realize (in our model) a matrix that is close to A. It is a natural question, how much the delivery time can be reduced by giving only an approximate representation of A satisfying certain minimum and maximum dose constraints.
Whereas the model with intensity matrices, 0-1-matrices as segments and all the problems described above lead to discrete (exact or approximate) decomposition problems, the last part of this thesis is devoted to a continuous segmentation approach that aims at finding a realistic model of fluence distributions. Because of scattering effects and the penumbra of radiation, the assumption that fluence distributions of shapes are 0-1-step functions does not hold in practice. In Section 7 the continuous model with real-valued target functions $f : [0,m] \times [0,n] \to \mathbb{R}_+$ and segments $S : [0,m] \times [0,n] \to [0,1]$ is introduced and similar approximation problems as in the discrete version are considered.
Both the discrete and the continuous approach have in common, that the output of the segmentation step is a set of MLC leaf positions, that serve as candidates for the final treatment plan. This set of candidates is enlarged by allowing all energies for each shape and then dose calculations using Monte-Carlo simulations are performed for all candidates [25]. The final treatment plan is then determined by solving the quadratic programming problem from [25] again. The output is a plan $(\mathcal{F}, \boldsymbol{x})$ that fixes a possible treatment for the patient.

3. SEGMENT CLASSES

To improve the readability of the thesis, we introduce a list of considered subsets of segments and explain, why they are of interest. Some of the constraints are required for technical limitations while others are needed due to dosimetric reasons. The 0-1-matrices satisfying the consecutive ones property (2.1) are called *segments* in general.

Tongue-and-groove constraint (TG)
To avoid radiation transmission through the small spaces between adjacent leaves of the MLC (interleaf leakage), the leaves are constructed such that they have a small overlap of the regions covered by adjacent leaves. This is indicated in Figure 3.1 and can lead to significant underdosage effects.

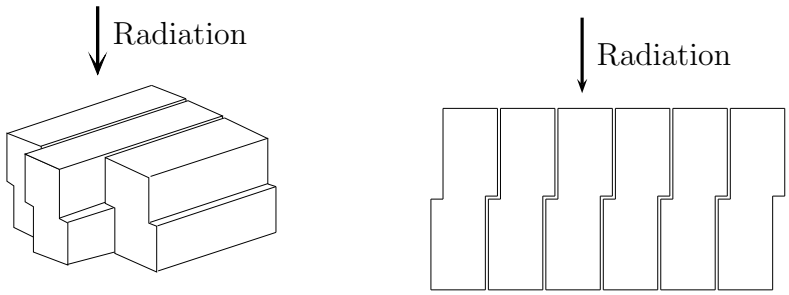

Fig. 3.1: The tongue-and-groove design of the leaves of an MLC.

The best we can achieve is that the overlap region between entry (i, j) and entry $(i+1, j)$ receives a fluence of $\min(a_{ij}, a_{i+1,j})$ by opening both bixels simultaneously as often as possible. Therefore, a segment S is called *TG-segment* if

$$\begin{aligned} a_{ij} \leq a_{i-1,j} \wedge s_{ij} = 1 \quad &\Longrightarrow \quad s_{i-1,j} = 1, \\ a_{ij} \geq a_{i-1,j} \wedge s_{i-1,j} = 1 \quad &\Longrightarrow \quad s_{ij} = 1 \end{aligned} \tag{3.1}$$

for $(i, j) \in [2, m] \times [n]$. Note that the definition depends on the input matrix A. The set of TG-segments is denoted by $\mathcal{S}^A_{TG}$.

Interleaf collision constraint (ICC)
For some MLCs (for instance the Elekta MLC) the left (respectively right) and right (respectively left) leaf of adjacent rows are not allowed to overlap because this would lead to a collision. Thus, a segment is called *ICC-segment* if

$$\ell_i \leq r_{i+1} + 1 \text{ and } r_i + 1 \geq \ell_{i+1} \text{ for all } i \in [m-1]. \tag{3.2}$$

Let $\mathcal{S}_{ICC}$ be the set of all ICC-segments.

Leaf overtravel constraint (LOC)
Let two parameters $b_\ell, b_r \in [n]$ with $b_\ell \geq b_r + 1$ be given. A segment is called *LOC-segment*, if

$$\ell_i \leq b_\ell \text{ and } r_i \geq b_r \text{ for all } i \in [m]. \tag{3.3}$$

In each row, the left leaf cannot be shifted more to the right than to the bixel with index b_ℓ and the right leaf cannot be shifted more to the left than to the bixel with index b_r. This is a technical constraint of some MLCs. The set of LOC-segments with respect to the parameters b_ℓ and b_r is called $\mathcal{S}_{LOC}^{b_\ell,b_r}$.

Minimum field size constraint (MFC)
For dosimetric reasons, the shapes of the segments should be sufficiently large and satisfy some minimum field size constraints. For larger segments the dose output can be calculated more accurately and scattering effects are smaller. We introduce three parameters $w \in [n]$, $h, f \in [m]$ with $h \leq f$. For simplicity of notation, we artificially enlarge the segments such that the row indices are $[-h+1, m+h]$ and the column indices are $[-w+1, n+w]$. For each segment S, we define $s_{ij} = 0$ for all $(i,j) \notin [m] \times [n]$. To simplify our considerations, we call rectangular bixel sets $[k,k'] \times [\ell,\ell'] \subseteq [-h+1, m+h] \times [-w+1, n+w]$ *rectangles* from now on. A rectangle spanning s rows and t columns is an $s \times t$*-rectangle*. The minimum field size is now represented by the following four requirements:

(i) Minimum Local Field Size: The ones of the segment can be covered by rectangles of ones of size $h \times w$, i.e. there exists a set of $h \times w$-rectangles in $[m] \times [n]$ such that $s_{ij} = 1$ for all the bixels in the union of the rectangles and $s_{ij} = 0$ otherwise.

(ii) Minimum Size of Closed Areas: The zeros of the segment can be covered by rectangles of zeros of size $h \times w$, i.e. there exists a set of $h \times w$-rectangles in $[-h+1, m+h] \times [-w+1, n+w]$ such that $s_{ij} = 0$ for all the bixels in the union of the rectangles and $s_{ij} = 1$ otherwise.

(iii) Row Overlap: If rows i and $i+1$ are not completely closed, we require $\min(r_i, r_{i+1}) - \max(\ell_i, \ell_{i+1}) \geq w - 1$, i.e. the openings of consecutive open rows should overlap in at least w bixels.

(iv) Minimum Total Field Height: At least f consecutive rows of the field are not totally closed, i.e. there are at least f consecutive rows with $\ell \leq r$. This ensures, that the total size of the MLC field is reasonably large.

A segment satisfying (i)-(iv) is called *MFC-segment*. The set of MFC-segments with respect to the parameters w, h and f is denoted by $\mathcal{S}_{MFC}^{w,h,f}$. The conditions (i) and (ii) make sure that we have no thin open regions and also no thin closed regions that are surrounded by open MLC bixels. Condition (iii) further relates to large connected areas, while condition (iv) stands for a large open region in total.

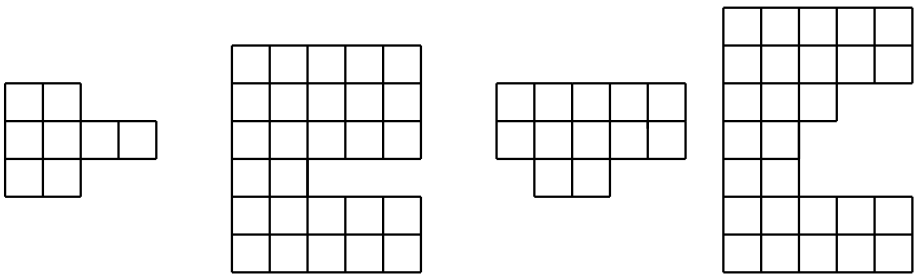

Fig. 3.2: Example for MFC-segments. For $w = h = 2$ and $f = 3$ the two left segments violate the minimum field size constraint, whereas the two right segments belong to $\mathcal{S}_{MFC}^{w,h,f}$.

Minimum separation constraint (MSC)
This constraint is a special case of the MFC, as we only require that each open row has a minimum leaf opening, i.e. for a given parameter $\lambda \in [n]$ (corresponding to w from above), a segment is called *MSC-segment*, if for each $i \in [m]$

$$r_i \geq \ell_i \Rightarrow r_i - \ell_i \geq \lambda - 1 \tag{3.4}$$

holds. The set of MSC segments with respect to λ is $\mathcal{S}_{MSC}^{\lambda}$.

If the value of the parameters and the matrix are clear from the context, we omit them and simply use $\mathcal{S}_{TG}$, $\mathcal{S}_{LOC}$, $\mathcal{S}_{MFC}$ and $\mathcal{S}_{MSC}$. For short, we also speak about TG-decompositions, if we mean decompositions into TG-segments and use analogous expressions for the other constraints.

As we will refer to special classes of segments later on in this thesis, we introduce some more notation. A segment is called *connected* if the irradiated area that corresponds to its leaf positions does not decompose into two or more parts, i.e. if the corresponding rectilinear polygon (considered as an open set) is connected. We denote by $\mathcal{S}_L$ the set of segments from $\mathcal{S}_{LOC} \cap \mathcal{S}_{MFC}$ that have connected open MLC regions. We call them L-segments (large segments), i.e. the set of connected MFC-segments satisfying the leaf overtravel constraint.

4. NEW ASPECTS ON EXACT SEGMENTATION

The exact decomposition problem **MIN-DT** has been studied extensively. If $\mathcal{S}' = \mathcal{S}$, i.e. all segments are allowed, the minimal DT equals [11, 24]

$$c(A) = \max_{i \in [m]} c_i(A) \quad \text{with} \quad c_i(A) = \sum_{j=1}^{n} (a_{ij} - a_{i,j-1})_+ \tag{4.1}$$

where $a_{i0} := 0$ for all $i \in [m]$. The problem is also solved for $\mathcal{S}' = \mathcal{S}_{ICC}$ [39] and $\mathcal{S}' = \mathcal{S}_{ICC} \cap \mathcal{S}_{TG}$ [41]. Engelbeen and Fiorini considered a further constraint called interleaf distance constraint (requiring that $|\ell_i - \ell_{i'}| \leq c$ for some given constant c and all rows i and i') and solved the problem under this constraint. Kamath *et al.* solved the problem for $\mathcal{S}' = \mathcal{S}_{TG}$, but only in the case when the leaves move from left to right from segment to segment, i.e. when we require unidirectional leaf movement. For the general case with $\mathcal{S}' = \mathcal{S}_{TG}$, no combinatorial polynomial time algorithm was found until now and from a theoretical point of view, the question for the computational complexity of the segmentation with tongue-and-groove constraint remains open. If one does not require the integrality of solutions, the problem is solvable in polynomial time, cf. [9]. Related problems dealing with the tongue-and-groove error can e.g. be found in [55].
For the sake of completeness, we mention a further exact decomposition problem, that can be studied. If the intensity matrix is too large to be delivered as a whole, field splitting algorithms that aim at minimizing the delivery time are used (cf. [16]).

We now introduce different approaches to the segmentation problem with $\mathcal{S}' = \mathcal{S}_{TG}$: two network flow formulations with side constraints and an efficient heuristics in Section 4.1 as well as a new integer linear programming (ILP) formulation in Section 4.2. But all exact approaches require solving ILPs and provide no polynomial time algorithm. Finally, a combinatorial polynomial time algorithm for the case that the input matrix A is binary, is explained in Section 4.3. We also show that determining the optimal delivery time in this case is equivalent to a coloring problem in a related perfect graph.

4.1 Network flow formulation for decompositions without tongue-and-groove underdosage

We consider the problem **MIN-DT** in the case where $\mathcal{S}' = \mathcal{S}_{TG}$. In [9] Boland *et al.* introduced a network flow formulation for decompositions into ICC-segments. This formulation can be modified to take tongue-and-groove underdosage effects into account. As just small changes in the networks have to be made, we only give the new

formulation and the results. The ideas of the proofs are the same as in [9]. In addition, we present an efficient heuristic approach to the problem.
Obviously, $c(A)$ is also a lower bound for the DT of segmentations satisfying the tongue-and-groove constraint. The following example shows that this bound is not sharp, in general.

Example 1. Consider the matrix

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

The minimal DT without tongue-and-groove constraint is 5, but it is a simple exercise to check that a DT of 6 is needed when the tongue-and-groove constraint is added to the model. The reason is that a segment with $\ell = 1$ and $r = 3$ in the first row is not allowed, because this leads to a conflict in row two. A possible segmentation of A with $DT = 6$ is

$$A = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{smallmatrix}\right) + 2 \cdot \left(\begin{smallmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{smallmatrix}\right) + \left(\begin{smallmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{smallmatrix}\right) + 2 \cdot \left(\begin{smallmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{smallmatrix}\right).$$

Now we show that optimal segmentations of A satisfying the tongue-and-groove constraint can be represented by minimum cost circulations in a suitably chosen network. As in [9], we define two different network flow problems: one with a simple graph and rather restrictive side constraints and one with a more complex graph with simple side constraints.

4.1.1 Two network flow formulations

Let us define a layered digraph $G = (V, E)$ as follows. G consists of m layers of nodes, in layer i the nodes represent the choice of the left and right leaf positions for row i ($i \in [m]$). This could be slightly ambiguous, since a completely covered row might be represented by any pair of leaf positions with $\ell = r + 1$. We resolve this by the convention that a completely closed row i is represented by the vertex $(i, 1, 0)$. Finally, we add two additional nodes D and D'. Thus, the set of nodes is

$$V \;=\; \{D, D'\} \cup \{(i, 1, 0) \;:\; i \in [m]\} \cup \{(i, \ell, r) \;:\; i \in [m],\, \ell \in [1, n],\, r \in [\ell, n]\}.$$

Two nodes in consecutive layers are connected by an arc if the corresponding leaf positions in the adjacent rows respect the tongue-and-groove constraint. If the leaf positions indicate, that a bixel (i, j) is irradiated while $(i \pm 1, j)$ is covered, the corresponding arc exists only if $a_{ij} > a_{i\pm1,j}$. More precisely, the choice of the leaf positions ℓ and r in row i and the leaf positions ℓ' and r' in row $i+1$ respects the tongue-and-groove constraint iff

$$\forall j \in [\ell, r] \setminus [\ell', r'] \quad a_{ij} > a_{i+1,j}, \text{ and} \tag{4.2}$$
$$\forall j \in [\ell', r'] \setminus [\ell, r] \quad a_{ij} < a_{i+1,j}. \tag{4.3}$$

We say that two nodes (i, ℓ, r) and $(i+1, \ell', r')$ *fit together* if (4.2) and (4.3) are satisfied.

Example 2. Consider the nodes $(i, 1, 4)$ and $(i+1, 3, 7)$ and the following two possibilities for the corresponding rows of A.

$$\begin{pmatrix} \mathbf{4} & \mathbf{4} & * & * & 2 & 1 & 5 \\ 3 & 2 & * & * & \mathbf{4} & \mathbf{4} & \mathbf{6} \end{pmatrix}, \qquad \begin{pmatrix} \mathbf{1} & \mathbf{4} & * & * & 2 & 1 & 5 \\ 3 & 2 & * & * & \mathbf{4} & \mathbf{4} & \mathbf{6} \end{pmatrix}$$

The bold faced entries indicate bixels receiving radiation. For the left matrix the two nodes fit together (and are thus joined by an arc) because of the inequalities $4 > 3$, $4 > 2$, $2 < 4$, $1 < 4$ and $5 < 6$. In contrast, for the right matrix the arc is missing since $1 \leq 3$.

Formally, we define the arc set of G as follows:

$$\begin{aligned} E = \{(D', D)\} &\cup \{(D, (1, \ell, r)) \ : \ (1, \ell, r) \in V\} \\ &\cup \{((m, \ell, r), D') \ : \ (m, \ell, r) \in V\} \\ &\cup \left\{((i, \ell, r), (i+1, \ell', r')) \ : \ \substack{(i,\ell,r),(i+1,\ell',r') \in V \\ (i,\ell,r) \text{ and } (i+1,\ell',r') \text{ fit together}}\right\}. \end{aligned}$$

As in [9], we have that $G \setminus \{D, D'\}$ is an acyclic digraph and there is a bijection between cycles in G and TG-segments. We illustrate this with an example.

Example 3. Figure 4.1 shows the graph G for the matrix $A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$. An optimal

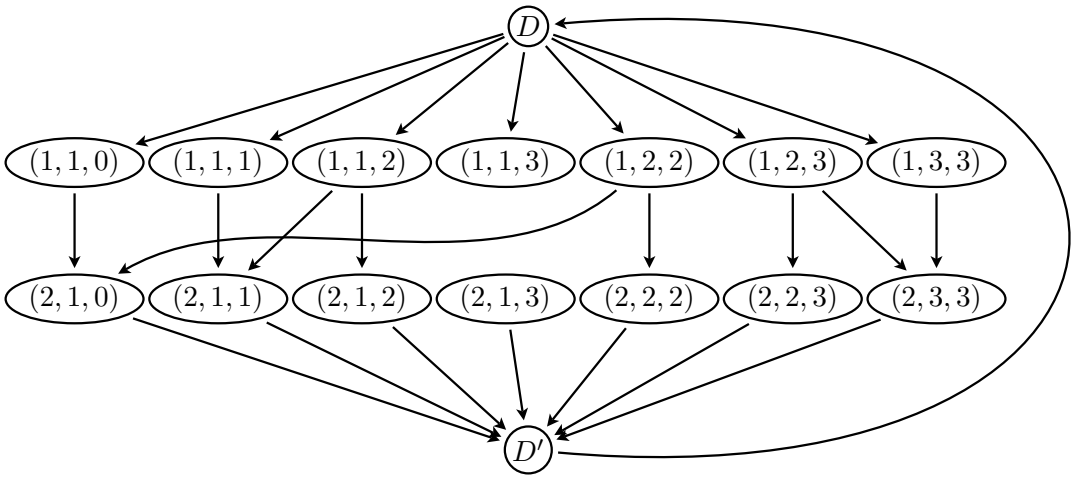

Fig. 4.1: The graph G for an example matrix.

segmentation is $A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ associated with the two cycles

$$(D, (1, 1, 2), (2, 1, 1), D', D) \quad \text{and} \quad (D, (1, 3, 3), (2, 3, 3), D', D).$$

As a consequence, associating a TG-segment S with a unit flow on the cycle $f(S)$, we obtain a correspondence between segmentations of A and integer circulations in G. A segmentation $A = \sum_{S \in \mathcal{S}_{TG}} u_S S$ corresponds to a circulation $\phi : E \to \mathbb{Z}_+$ with

$$\phi(D', D) = \sum_{S \in \mathcal{S}_{TG}} u_S = DT.$$

Hence, in order to solve the segmentation problem we should minimize the cost of ϕ with respect to the cost function

$$c_e = \begin{cases} 1 \text{ if } e = (D', D), \\ 0 \text{ otherwise.} \end{cases}$$

Unfortunately, not every circulation is associated with a segmentation. It is necessary and sufficient, that the total flow through all nodes corresponding to leaf positions with bixel (i, j) open equals a_{ij}. More precisely, we have to impose the side constraints

$$\sum_{\substack{(i,\ell,r)\in V \\ \ell\le j\le r}} \sum_{q:(q,(i,\ell,r))\in E} \phi(q, (i, \ell, r)) = a_{ij} \quad ((i, j) \in [m] \times [n]). \tag{4.4}$$

In conclusion, we are prepared to formulate the main result of this section (compare the corresponding statement in [9]).

Theorem 1. *Finding a minimum cost circulation in the network $G = (V, E)$ with the side constraints (4.4) solves the problem **MIN-DT** for $\mathcal{S}' = \mathcal{S}_{TG}$.*

As in [9], we extend these ideas and formulate another network flow formulation with simpler side constraints. A second layered digraph $G' = (V', E')$ is defined as follows. G' has $3m$ layers of nodes where every 3 consecutive layers form a block and correspond to a row i of the intensity matrix. Let us describe the types of nodes in such a block for row i in more detail.

- In the first and third layer of the block, we find nodes $(i, \ell, r)^t$ for $t = 1, 2$ representing the possible leaf positions in row i, as before.

- The second layer of the block contains nodes (i, j) with $j \in [0, n]$.

The idea is that the flow enters a block via a node of the form $(i, \ell, r)^1$, runs through some arcs in layer two of the block and finally leaves the block via $(i, \ell, r)^2$. In layer two the flow goes through arcs representing bixels that are exposed to radiation if leaf positions ℓ and r are chosen in row i. To summarize, we get the set of nodes $V' = V'_1 \cup V'_2 \cup V'_3$ with

$$\begin{aligned} V'_1 &= \{D, D'\}, \\ V'_2 &= \left\{(i, \ell, r)^1,\ (i, \ell, r)^2 \ :\ i \in [m],\ \ell \in [1, n],\ r \in [\ell, n]\right\} \\ &\qquad \cup \left\{(i, 1, 0)^1, (i, 1, 0)^2 \ :\ i \in [m]\right\}, \\ V'_3 &= \left\{(i, j) \ :\ i \in [m],\ j \in [0, n]\right\}. \end{aligned}$$

The set of arcs is $E' = E'_1 \cup E'_2 \cup E'_3 \cup E'_4$ with

$$\begin{aligned}
E'_1 &= \{(D', D)\} \cup \left\{\left(D, (1,\ell,r)^1\right) \ : \ (1,\ell,r)^1 \in V'_2\right\} \\
&\qquad \cup \left\{\left((m,\ell,r)^2, D'\right) \ : \ (m,\ell,r)^2 \in V'_2\right\}, \\
E'_2 &= \left\{\left((i,\ell,r)^1, (i,\ell-1)\right) \ : \ (i,\ell,r)^1 \in V'_2\right\} \\
&\qquad \cup \left\{\left((i,r), (i,\ell,r)^2\right) \ : \ (i,\ell,r)^2 \in V'_2\right\}, \\
E'_3 &= \left\{\left((i,\ell,r)^2, (i+1,\ell',r')^1\right) \ : \ \begin{smallmatrix}(i,\ell,r)^2,\ (i+1,\ell',r')^1 \in V'_2 \\ (i,\ell,r) \text{ and } (i+1,\ell',r') \text{ fit together}\end{smallmatrix}\right\}, \\
E'_4 &= \{((i,j-1),(i,j)) \ : \ i \in [m],\ j \in [n]\}.
\end{aligned}$$

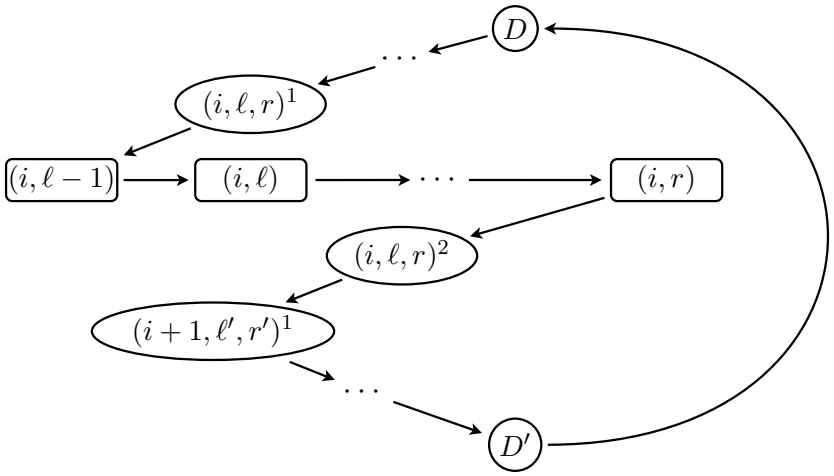

Fig. 4.2: Cycle in the graph G'.

Figure 4.2 illustrates the construction. As before, we want to identify segmentations with circulations in G'. We need an additional constraint.

Definition 1 (Matched circulation)**.** A circulation $\phi' : E' \to \mathbb{Z}_+$ is called *matched*, if

$$\phi'((i,\ell,r)^1, (i,\ell-1)) = \phi'((i,r), (i,\ell,r)^2) \tag{4.5}$$

for all $i \in [m]$ and all suitable pairs (ℓ, r).

Again, we define the arc cost $c_{(D',D)} = 1$ and the costs of all the other arcs are set to 0. Finally, we define lower and upper capacities $\underline{u}_e$ and $\overline{u}_e$ of the arcs by

$$\begin{aligned}
&\underline{u}_e = 0,\ \overline{u}_e = \infty && \text{for } e \in E'_1 \cup E'_2 \cup E'_3, \\
&\underline{u}_e = \overline{u}_e = a_{ij} && \text{for } e = ((i,j-1),(i,j)) \in E'_4.
\end{aligned} \tag{4.6}$$

Intuitively, the flow on arc $((i,j-1),(i,j))$ represents the total fluence in bixel (i,j), so the capacities (4.6) ensure the delivery of the correct fluence. As in [9], it can be shown that the segmentation problem with minimal DT is equivalent to the following network flow problem:

$$\text{Find a circulation } \phi' \text{ with } \phi'(D', D) \to \min$$

subject to ϕ' satisfying the lower and upper capacities (4.6) and the side constraints (4.5). This result can be proved directly or as a consequence of the following theorem.

Theorem 2. *There is a cost preserving bijection between flows ϕ in G satisfying the side constraints (4.4) and matched circulations ϕ' in G' satisfying the capacity constraints (4.6).*

Again, we omit the proof, as it is exactly the same as the proof for Theorem 4.4 in [9] for the segmentation with interleaf collision constraint. Theorem 2 shows that we can solve the minimum delivery time problem without tongue-and-groove underdosage as follows.

- Find a flow ϕ' with minimal cost $\phi'(D', D)$ in G' subject to the capacities (4.6) and the side constraints (4.5).
- Determine the corresponding flow ϕ in G.
- Decompose ϕ into $\phi(D', D)$ cyclic unit flows, each corresponding to a TG-segment.

In conclusion, we have reduced the segmentation problem to a minimum cost flow problem with side constraints in a network with $O(mn^2)$ nodes and $O(mn^4)$ arcs. Thus, the corresponding ILP has $O(mn^2)$ constraints and $O(mn^4)$ variables.

4.1.2 Heuristic segmentation algorithm

Solving the integer linear programs that result from our two network flow formulations is rather time-consuming. Therefore, we use the graph $G = (V, E)$ of our first network flow formulation to heuristically compute a segmentation with a small delivery time. Recall that, for $i \in [m]$, the i-th row complexity is defined as

$$c_i(A) = \sum_{j=1}^{n} (a_{ij} - a_{i,j-1})_+$$

with $a_{i0} = 0$. Let $c'_i(\ell, r)$ denote the row complexity of row i after the subtraction of a segment with $\ell_i = \ell$ and $r_i = r$. More formally (using $a_{i0} = a_{i,n+1} = 0$),

$$c'_i(\ell, r) = \begin{cases} c_i(A) + 1 & \text{if } \ell \le r,\ a_{i\ell} \le a_{i,\ell-1} \text{ and } a_{ir} \le a_{i,r+1}, \\ c_i(A) & \text{if } \ell \le r,\ a_{i\ell} \le a_{i,\ell-1} \text{ and } a_{ir} > a_{i,r+1}, \\ c_i(A) & \text{if } \ell \le r,\ a_{i\ell} > a_{i,\ell-1} \text{ and } a_{ir} \le a_{i,r+1}, \\ c_i(A) & \text{if } \ell = r + 1, \\ c_i(A) - 1 & \text{if } \ell \le r,\ a_{i\ell} > a_{i,\ell-1} \text{ and } a_{ir} > a_{i,r+1}. \end{cases}$$

We define the weight of an arc $e = (v, (i, \ell, r))$ with endnode (i, ℓ, r) by

$$w(e) = \begin{cases} 1 & \text{if } c'_i(\ell, r) \ge c(A), \\ 0 & \text{otherwise.} \end{cases}$$

The arcs with endnode D' have weight 0. We can determine a segmentation without tongue-and-groove underdosage as described in Algorithm 1.

Algorithm 1 Heuristic segmentation

Input: matrix A

$\mathcal{S} = \emptyset$

while $A \neq \mathbf{0}$ **do**

Determine row complexities, arcs and weights with respect to A.

Compute a shortest D-D'-path $(D, (1, \ell_1, r_1), \ldots, (m, \ell_m, r_m), D')$.

Let S be the corresponding segment, i.e. $s_{ij} = \begin{cases} 1 & \text{if } \ell_i \leq j \leq r_i, \\ 0 & \text{otherwise.} \end{cases}$

$\mathcal{S} = \mathcal{S} \cup \{S\}$

$A = A - S$

end while

Output: segmentation $\mathcal{S}$

If there are several choices for the pair (ℓ_i, r_i) in some row, we choose one with maximal difference $r_i - \ell_i$. This heuristics is guided by the lower bound $c(A)$: in each step we try to find a segment S such that $c(A - S) = c(A) - 1$.
Computational results show that the heuristics very often finds the optimal solution and always a solution close to the optimum. The computation times for the heuristics are significantly smaller than for solving the ILP (on a 2.5GHz workstation). We implemented our heuristic method, the heuristics of Kamath and our ILP formulation in C++. The integer linear program was solved using Gurobi [59]. We tested the approaches on 475 clinical intensity matrices provided by the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf. Column "$c(A)$" in Table 4.1 contains the optimal DT for unconstrained decomposition, "Kamath" is the optimal DT for unidirectional leaf sequences according to [49], "heuristic" is the DT from our heuristics and "exact" is the DT for the segmentation corresponding to the solution of the constrained network flow formulation.

m	n	$c(A)$	Kamath	heuristic	exact
19.47	20.76	39.41	46.56	42.65	41.08

Tab. 4.1: Average test results for TG-segmentations.

0	1	2	3	4	5	6	7	> 7
178	115	71	53	16	17	11	4	10

Tab. 4.2: Frequencies of the differences between the heuristic DT and the exact minimum.

max	total
47 minutes	7 hours

Tab. 4.3: Computation times for the exact ILP solution.

Table 4.2 shows the frequencies of the differences between the exact and the heuristic solutions. The first line of the table contains the differences and the second line the number of instances, where the difference occurred. The computation time for the heuristics is completely acceptable for practical purposes (approximately a second for each matrix). For the exact solution, computation time becomes an issue, as indicated

by Table 4.3, which shows the maximal computation time for a single matrix, and the total time for the segmentation of all 475 matrices.

4.2 An integer linear programming formulation for decompositions without tongue-and-groove underdosage

As an alternative to the network flow approach in Section 4.1 we now introduce another integer linear programming problem for solving the decomposition problem without tongue-and-groove underdosage. It turns out that this formulation needs less variables and constraints than the ILP formulation that corresponds to the network if the minimal delivery time of a decomposition into TG-segments is small.
Let $T \in \mathbb{N}$ be an upper bound for the delivery time. We want to compute a TG-decomposition $A = S^1 + \cdots + S^T$. We modify an approach from [9] and introduce binary variables as follows:

$$z_t = \begin{cases} 1, & \text{if } S^t \neq \mathbf{0} \\ 0, & \text{otherwise} \end{cases} \qquad t \in [T], \tag{4.7}$$

$$y_{ijt} = \begin{cases} 1, & \text{if } S^t_{ij} = 1 \\ 0, & \text{otherwise} \end{cases} \qquad t \in [T], (i,j) \in [m] \times [n], \tag{4.8}$$

$$L_{ijt} = \begin{cases} 1, & \text{if } \ell^t_i = j \\ 0, & \text{otherwise} \end{cases} \qquad t \in [T], (i,j) \in [m] \times [n], \tag{4.9}$$

$$R_{ijt} = \begin{cases} 1, & \text{if } r^t_i = j \\ 0, & \text{otherwise} \end{cases} \qquad t \in [T], (i,j) \in [m] \times [n]. \tag{4.10}$$

Let M be a sufficiently large natural number, e.g. $M = mn$. Our ILP then reads as follows:

$$z_t \geq z_{t+1} \qquad t \in [T-1], \tag{4.11}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} y_{ijt} \leq M z_t \qquad t \in [T], \tag{4.12}$$

$$\sum_{t=1}^{T} y_{ijt} = a_{ij} \qquad (i,j) \in [m] \times [n], \tag{4.13}$$

$$\sum_{j=1}^{n} L_{ijt} = 1 \qquad i \in [m], t \in [T], \tag{4.14}$$

$$\sum_{j=1}^{n} R_{ijt} = 1 \qquad i \in [m], t \in [T], \tag{4.15}$$

$$\sum_{\ell=1}^{j} L_{i\ell t} - \sum_{r=1}^{j-1} R_{irt} = y_{ijt} \qquad (i,j) \in [m] \times [n], t \in [T], \tag{4.16}$$

$$y_{ijt} \geq y_{i+1,j,t} \qquad (i,j) \in [m-1] \times [n], t \in [T], a_{ij} \geq a_{i+1,j}, \tag{4.17}$$

$$y_{ijt} \leq y_{i+1,j,t} \qquad (i,j) \in [m-1] \times [n], t \in [T], a_{ij} \leq a_{i+1,j}, \tag{4.18}$$

$$z_t, y_{ijt}, L_{ijt}, R_{ijt} \in \{0,1\} \qquad (i,j) \in [m] \times [n], t \in [T], \tag{4.19}$$

$$\sum_{t=1}^{T} z_t \to \min. \tag{4.20}$$

Inequality (4.11) ensures that the first segments are the nonzero ones and the possibly empty segments follow at the end. Inequality (4.12) makes sure that $z_t = 1$ if the corresponding segment is nonzero. The segmentation property is (4.13). In each row of a segment, we need exactly one position of the left leaf and one position of the right leaf which is modeled by equations (4.14) and (4.15). If $y_{ijt} = 1$ then $L_{i\ell t} = 1$ for some $\ell \leq j$ and $R_{irt} = 1$ for some $r \geq j$ is required. If $y_{ijt} = 0$ and $L_{i\ell t} = 1$ for some $\ell \leq j$, then R_{irt} must be equal to 1 for some $r \leq j-1$ to close bixel (i,j). If $y_{ijt} = 0$ and $L_{i\ell t} = 1$ for some $\ell > j$, then one can choose an index $r \geq j$ with $R_{irt} = 1$ to close bixel (i,j). This is encoded by (4.16) which satisfies the consecutive ones property in each row. Inequalities (4.17) and (4.18) are the tongue-and-groove constraints. Finally, the objective function (4.20) is the number of nonzero segments which is the delivery time. Note that some segments can occur several times for different indices t. This ILP formulation has $O(mnT)$ variables and $O(mnT)$ constraints and might beat the approach from Section 4.1 if T is not too large, which is the case if the entries of A are small and if there are not too many large positive differences $(a_{ij} - a_{i-1,j})_+$ in the matrix. But experimental tests show that for common intensity matrices, the network flow formulations mostly outperform this ILP formulation.

4.3 Decompositions without tongue-and-groove underdosage for binary input matrices

In this section, we restrict ourselves to decompositions of binary fluence matrices. Let $A = (a_{ij})$ denote the given binary fluence matrix of size $m \times n$. Obviously, we have $u_S = 1$ for all TG-segments S arising in the decomposition. Thus, minimizing the delivery time is equivalent to minimizing the number of segments. First, we prove that the problem **MIN-DT** for $\mathcal{S}' = \mathcal{S}_{TG}$ and binary fluence matrices can be solved in polynomial time and provide an $O(m^2n^2)$ time algorithm to find such an optimal TG-decomposition. Then we show that finding the optimal delivery time of a TG-decomposition (but not the decomposition itself) can be done by relating the problem to a coloring problem in a perfect graph, which gives an alternative proof for the polynomiality of the problem.

4.3.1 Polynomial algorithm

For simplicity of notation, we add a 0-th and an $(n+1)$-th column to A and put $a_{i0} = a_{i,n+1} = 0$ for all $i \in [m]$. Similarly, we add a 0-th and an $(m+1)$-th row to A and put $a_{0j} = a_{m+1,j} = 0$ for all $j \in [0, n+1]$. Obviously, if the input matrix is binary, the tongue-and-groove-constraint reduces to the fact, that consecutive ones in a column have to be irradiated simultaneously.

Definition 2 (Box)**.** For $i_1, i_2 \in [m]$, $i_1 \leq i_2$ and $j \in [n]$, the set of bixels $B = \{(i,j) \mid i_1 \leq i \leq i_2\}$ is called a *box*, if $a_{ij} = 1$ for all $(i,j) \in B$ and $a_{i_1-1,j} = a_{i_2+1,j} = 0$. If $a_{ij} = 1$ for $(i,j) \in [m] \times [n]$, we denote the unique box containing (i,j) by B_{ij}. If $a_{ij} = 0$, we define $B_{ij} = \emptyset$. Let the set of all boxes be $\mathcal{B}$.

In the unconstrained case, a decomposition of a binary matrix corresponds to a partition of the set of ones such that each subset forms a segment. Including the tongue-and-groove constraint, a decomposition is a partition of the set of boxes such that each subset has the consecutive ones property. Figure 4.3 shows an example for a decomposition of the set of boxes into MLC-segments.

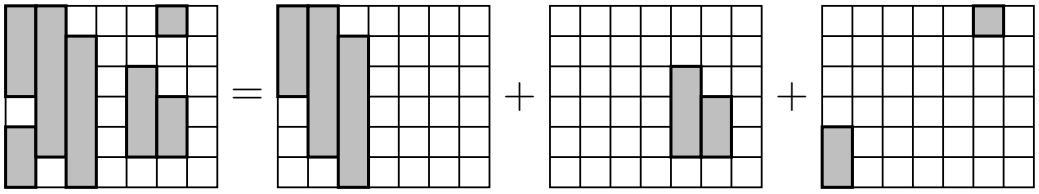

Fig. 4.3: Decomposition of the boxes into TG-segments.

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

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

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

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

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

Definition 3 (Boxes of row i and number of splits)**.** Let $i \in [m]$ be fixed. The *boxes of row i* are the elements of the set $\{B_{ij} \mid j \in [n]\}$. The *number of splits of row i* is defined as the minimal number of splits between neighboring boxes of row i that are necessary to make all the connected subsets of $\bigcup_{j \in [n]} B_{ij}$ satisfy the consecutive ones property. This number is denoted by $s_i(A)$.

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

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

for some $i' < i$ or

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

for some $i' > i$. We call the set $\{B_{i,j}, \ldots, B_{i,j'}\}$ an *i-cup* in the first case and an *i-cap* in the second case, as the zeros can be crossed below or above via other rows of ones. The situation is illustrated in Figure 4.4. If we talk about i-caps or i-cups, we call them i-obstacles. The comprised zero entries $a_{i',\ell}$ for $j+1 \leq \ell \leq j'-1$ are called *critical zeros*, as they destroy the consecutive ones property of the corresponding boxes and imply the necessity of a split.

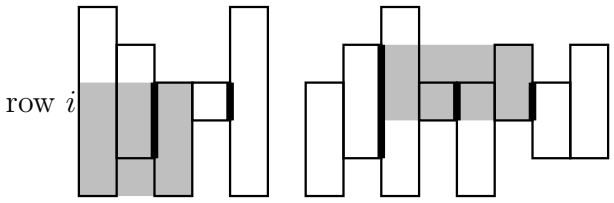

Fig. 4.4: Number of splits in row $i \in [m]$. The grey areas indicate an i-cap (left) and an i-cup (right). The thick lines indicate the splits that destroy the i-obstacles. Sometimes, one split can destroy two i-obstacles (one i-cap and one i-cup) as indicated by the second split on the right.

For each row $i \in [m]$ and each i-obstacle $\{B_{ij}, \ldots, B_{ij'}\}$, we get an integral interval of possible split positions $[j, j'-1]$. At least one of these splits has to be chosen in order to destroy the i-obstacle and make the connected boxes satisfy the consecutive ones property. Let therefore $K_1^i = [k_1, k_1'-1]^i, K_2^i = [k_2, k_2'-1]^i, \ldots, K_{v_i}^i = [k_{v_i}, k_{v_i}'-1]^i$ be the integral split intervals for all i-cups (ordered from left to right) and analogously let $L_1^i = [\ell_1, \ell_1'-1]^i, L_2^i = [\ell_2, \ell_2'-1]^i, \ldots, L_{w_i}^i = [\ell_{w_i}, \ell_{w_i}'-1]^i$ be the integral split intervals for all i-caps (ordered from left to right). Here, v_i is the number of i-cups and w_i is the number of i-caps. Obviously, the K_j^i are pairwise disjoint and the L_j^i are pairwise disjoint for fixed i. Thus, for all possible split positions $j \in [n-1]$, j can be contained in at most two of the intervals from above. As $s_i(A)$ is the minimal number of splits needed to destroy all i-obstacles, the computation of $s_i(A)$ amounts to finding a subset $M \subseteq [n-1]$ such that:

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

The optimal value of the objective function is $s_i(A)$. This problem aims at partitioning the vertices of the corresponding interval graph into a minimum number of cliques (for more details see e.g. [38]). It can easily be solved by taking all the intervals of a row from the left to the right, and insert a split in the last possible position, that is the last position for which otherwise there would be an unsplit interval. Figure 4.5 gives a possible optimal solution for the problem, where the arrows indicate the splits given by this procedure. The first row in Figure 4.5 represents all the split intervals corresponding to the i-cups and the second row represents all the split intervals corresponding to the i-caps.

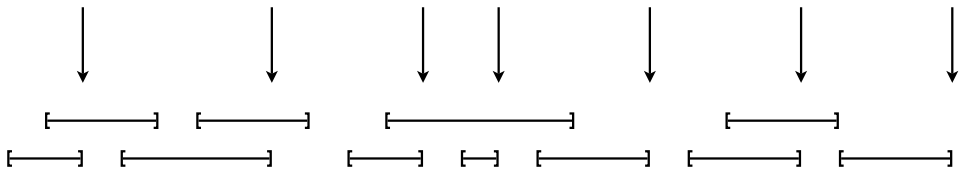

Fig. 4.5: Possible splits.

Definition 4 (Tongue-and-groove complexity)**.** We define the *TG row complexity* of row i by

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

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

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

Recall that, in the binary case, $c_i(A)$ is the number of columns j such that $a_{ij} = 1$ and $a_{i,j-1} = 0$ (using $a_{i0} = 0$). Our aim is to show that $c^{TG}(A)$ is the minimal delivery time of a segmentation of the binary matrix A into TG-segments. For this, we need some more notation and lemmas. Obviously, $c_i^{TG}(A)$ is the minimal number of TG-segments we need to decompose the boxes of row i and $c^{TG}(A)$ is a lower bound for the minimal delivery time.
Let us now assume that we have given A together with a set of splits $\mathcal{SP}$. If there exists some $s \in \mathcal{SP}$ with $(i, j) \in s$, then we do not allow to put the bixels (i, j) and $(i, j + 1)$ into the same segment. We now generalize the definition of $c_i^{TG}(A, \mathcal{SP})$ and define it as the minimum number of segments that are necessary to decompose the set of boxes of row i with respect to the split set $\mathcal{SP}$. Obviously, if $\mathcal{SP} = \emptyset$, this corresponds to our previous definition of $c_i^{TG}(A)$. The i-obstacles and the corresponding split intervals for all $i \in [m]$ are also defined with respect to $\mathcal{SP}$, i.e. including splits reduces the number of i-obstacles. If we insert a split between neighboring boxes B and B' and the split affects row i, there are two cases:

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

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

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

we have

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

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

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

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

The next lemma follows from the previous one. We call it sharing lemma as we will refer to it several times.

Lemma 2. *(Sharing lemma)*

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

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

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

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

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

Before we can prove the next lemma, we need some more notation. To clarify the notions, we introduce here the notations in terms of the split intervals as well as in terms of interval graphs. So, let the split intervals of some row $i \in [m]$ be ordered such that for consecutive intervals $I = [i_1, i_2]$ and $J = [j_1, j_2]$ $i_1 \leq j_1$ and if $i_1 = j_1$ then $i_2 \geq j_2$ holds. This means, if two intervals start at the same position, the longer one comes first with respect to this order. We associate with these intervals the interval graph $G_i(V, E)$. The set of vertices V includes a vertex I for each split interval of the row i and two vertices I and J are connected if the two corresponding split intervals have a non-empty intersection. Let us notice that $G_i(V, E)$ is a forest.
A set of consecutive split intervals $(I_1, \ldots, I_k)$ forms a *sequence* of row i if $I = I_1 \cup \cdots \cup I_k$ is a connected interval. The corresponding vertices $I_1, \ldots, I_k$ form a set of connected vertices of G_i. Such a set is a *component* if it cannot be extended, which means that $I_1, \ldots, I_k$ form a connected component of G_i. Finally, a *trunk* is a maximal sequence $(I_1, \ldots, I_{\tilde{k}})$ with $\tilde{k} \leq k$ that has the property that there is no interval in the component $(I_1, \ldots, I_k)$ that is contained in any of the intervals $I_1, \ldots, I_{\tilde{k}}$. Note that a trunk consists of a set of split intervals corresponding to alternating i-caps and i-cups. The definitions are illustrated in Figure 4.6 and 4.7.

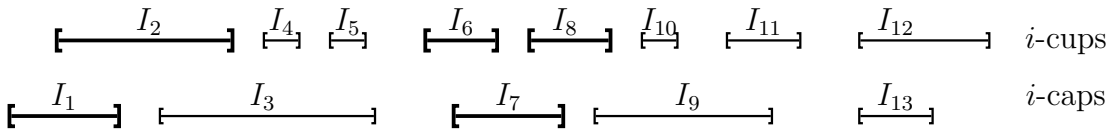

Fig. 4.6: Components and trunks. The intervals $I_1, \ldots, I_{13}$ are the split intervals of some row i. They decompose into three components of split intervals $(I_1, \ldots, I_5), (I_6, \ldots, I_{11})$ and (I_{12}, I_{13}). The trunks (I_1, I_2) and (I_6, I_7, I_8) are highlighted with bold lines. The trunk of the last component is empty.

Obviously, for split intervals I and J, if $J \subseteq I$ then every split in J automatically also splits I. Thus, for a component $(I_1, \ldots, I_k)$ in row i with I_1 starting left from I_2, the decision if a split in $I_1 \setminus I_2$ is i-feasible only depends on the trunk. More detailed, if the trunk contains t split intervals, we always need $\lceil t/2 \rceil$ splits to destroy all of them. If t is even, each split has to destroy two consecutive intervals, while for odd t one split may only destroy the first interval. Hence, for the first component in Figure 4.6, we see that each split of I_4 and I_5 will also split I_3. Therefore, because I_3 will automatically

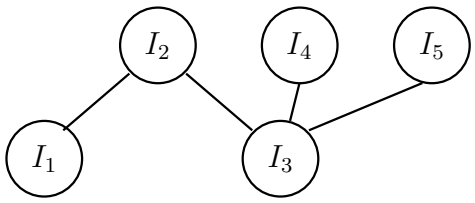

Fig. 4.7: The first component of the interval graph corresponding to the intervals $I_1, \ldots, I_5$ from Figure 4.6.

be split by the split we will have to insert in I_4, we do not have to care of that interval and the decision about the feasibility of a split in $I_1 \setminus I_2$ only depends on the trunk (I_1, I_2). As the number of intervals in this trunk is even, a split in $I_1 \setminus I_2$ is infeasible. The next lemma is obvious using the interval graphs.

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

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

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

We propose the following

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

Obviously, at the end there are exactly $s_i(A)$ splits and $c_i^{TG}(A)$ connected regions of ones in each row $i \in [m]$.
The only thing we still have to prove is that the choice of a feasible split in the splitting procedure is always possible.

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

Proof. As there is a connected region of ones that does not satisfy the consecutive ones property, there exists an i-cap or an i-cup for some row $i \in [m]$. We consider the i-obstacle with the leftmost critical zero and under this circumstance minimal value of i.
We can again w.l.o.g. assume that this is a subset of ones of the form of an i-cup, because the case of an i-cap is similar. Let the leftmost split interval in row i be $[j, j' - 1]$ and no split of type $s_{B_{ik}, B_{i,k+1}}$ with $k \in [j, j' - 1]$ is already in $\mathcal{SP}$. Let i' be the last row below row i, for which this is also an i'-cup. Possibly, $i' = i$. Since $[j, j' - 1]$

is the split interval of the leftmost i-cup we know that all the nonempty trunks which contain $[j, j'-1]$ start in that split interval. We have the following situation where at least one of the $*$-positions is a 0:

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

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

Case 1: For all $k \in [i, i']$, there is no k-cap with split interval $[\ell, \ell'-1]$ such that $[j, j'-1] \cap [\ell, \ell'-1] \neq \emptyset$. As there is at least one zero in row $i'+1$ in $[j, j']$, there is a split that only affects rows $k \in [i, i']$ for which there is the k-cup with split interval $[j, j'-1]$. Using Lemma 3 b) we obtain the feasibility of this split.

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

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

Let us assume that the splits in $[j, j'-1] \cap [\ell, \ell'-1]$ are k-infeasible for some row $k \in [i'+1, k_2]$ (if not, there is a feasible split). That means, in row k it is not allowed to split only the k-cap in $[\ell, \ell'-1]$, because it has to be cut together with a k-cup on the right (cf. Figure 4.8). Lemma 3 c) tells us that the trunk of split intervals in row k starting with $[\ell, \ell'-1]$ ends with a split interval corresponding to a k-cup (because the total number of vertices in that trunk must be even). Thus the trunk of split intervals in row k is of the form $[\ell, \ell'-1] = J_1, I_1, J_2, I_2, \dots, J_t, I_t$, where the $J_1, \dots, J_t$ are k-caps and the $I_1, \dots, I_t$ are k-cups. Now we use Lemma 2 several times: Because row k and the rows in $[k_1, i']$ share J_1, the sharing lemma (Lemma 2) tells us that they also share I_1. Now there are two parts: For the rows in $h \in [k_1, i']$ which do not share J_2, the trunk of these rows starts and ends with a cup (and thus consists of an odd number of intervals) and thus every split in $[j, j'-1] \setminus [\ell, \ell'-1]$ is h-feasible. And for the rows in $h \in [k_1, i']$ which share J_2, we use the sharing lemma again and we obtain, that they share I_2 and so on. Thus, for all $h \in [k_1, i']$, we either

find an h-feasible split in $[j, j'-1] \setminus [\ell, \ell'-1]$ (*) (because the trunk has an odd number of split intervals) or row h shares all the split intervals with row k (**). Furthermore, the trunks of split intervals in rows $h \in [k_1, i']$ cannot be longer than $([j, j'-1], J_1, I_1, J_2, I_2, \ldots, J_t, I_t)$, as an i'-cap would have to follow that, again using Lemma 2 would also be a k-cap, a contradiction. Thus, the trunk for the rows $h \in [k_1, i']$ is exactly $([j, j'-1], J_1, I_1, J_2, I_2, \ldots, J_t, I_t)$ in case (**) and the number of split intervals is again odd. Again, every split in $[j, j'-1] \setminus [\ell, \ell'-1]$ is h-feasible. All in all, every split in $[j, j'-1] \setminus [\ell, \ell'-1]$ is feasible, as it is h-feasible for all the split rows h.

□

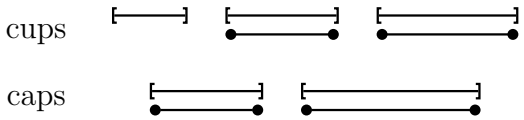

Fig. 4.8: Trunks of split intervals. The intervals with the circles at the end are the k-obstacles, the others those of the rows in $[k_1, i']$.

The result of our splitting procedure is the following: We have inserted a number of feasible splits, until no more feasible splits are possible. Afterwards each row i is split exactly $s_i(A)$ times and all the connected regions of ones in the matrix have the consecutive ones property. We have $c_i^{TG}(A)$ connected regions of ones intersecting with row i for all $i \in [m]$. The splitting procedure takes time $O(m^2n^2)$. At first, in each row $i \in [m]$ and for each block B_{ij} we need at most mn operations to check, if a split after this block is necessary. Thus, it takes $O(m^2n^2)$ operations to find all split intervals for all rows. With these split intervals it takes time $O(n)$ to find $s_i(A)$ in each row i. Afterwards, checking that a split is i-feasible can be done in time $O(n)$ by computing the minimal number of splits for the left part and for the right part.
We will now define a step of the segmentation procedure, that finds for given A a TG-segment S such that $A-S$ is nonnegative and $c^{TG}(A-S) = c^{TG}(A)-1$. Let us assume that we have already obtained the set $\mathcal{SP}$ of splits from the splitting procedure, i.e. we have a number of connected regions of ones with consecutive ones property, whose union is the set of ones in A. We call a row $i \in [m]$ *critical* if $c_i^{TG}(A) = c^{TG}(A)$. For $i \in [m]$ let $\boldsymbol{s}_i$ denote the i-th row of S.
The segmentation step is explained in Algorithm 2. We prove in Lemma 5 that we can always find such a region for each critical row, which is still empty in S. Because the segmentation procedure selects only connected regions of ones from A it obviously follows that $A - S$ is nonnegative. Moreover, because all critical rows i satisfy $\boldsymbol{s}_i \neq \mathbf{0}$ at the end of the for-loop, we also have that $c^{TG}(A - S) = c^{TG}(A) - 1$. Hence the segmentation procedure will lead us to a segmentation of A which uses $c^{TG}(A)$ TG-segments, when we iterate it until $A = \mathbf{0}$. The only thing we still have to check is the fact that for each critical row i such that still $\boldsymbol{s}_i = \mathbf{0}$ in the for-loop, we can always find a connected region of ones with $\boldsymbol{s}_i \neq \mathbf{0}$ that does not intersect a non-empty row in the current segment S (see Figure 4.9) and that can be added to S.

Algorithm 2 Segmentation

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

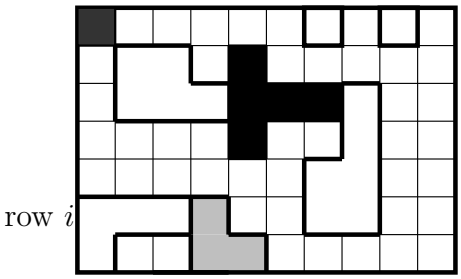

Fig. 4.9: Example for the segmentation procedure. A matrix A with its connected regions of ones with respect to the splits from the splitting procedure. The black areas form the current segment S in the for-loop of the segmentation algorithm. The first critical row which is still empty in S is row $i = 6$. The grey area intersects with row $i = 6$ but does not intersect with non-empty rows of S. So, we can choose this connected region of ones to complete S.

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

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

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

Theorem 3. *The minimal delivery time of a segmentation of A into TG-segments is $c^{TG}(A)$.*

Proof. It is obvious that we need at least $c^{TG}(A)$ TG-segments to decompose A because there is some row i^* whose boxes can only be decomposed by at least $c^{TG}_{i^*}(A) = c^{TG}(A)$ segments. After the splitting procedure, we find at most $c^{TG}(A)$ regions of ones in each row $i \in [m]$ and eliminating one of them always corresponds to decreasing the TG row complexity of row i by 1. Obviously, Algorithm 2 finds a TG-segment that decreases the TG row complexity by 1 in all the critical rows (and maybe also in some other rows). The statement then follows by induction. □

Corollary 4. *The optimal decomposition of a binary input matrix into TG-segments can be found in polynomial time.*

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

We close this section with an

Example 4. We discuss our whole approach using the example matrix

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1| & 1 & 1 & 1| & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1| & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } S_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

from Figure 4.9 with the splits indicated by vertical bars. The first segment S_1 is determined after computing the TG-row-complexities

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

deducing $c^{TG}(A) = 3$, applying the splitting procedure and the first step of the segmentation. After inserting feasible splits, the connected regions of ones are according to Figure 4.9. In the first step of the segmentation, the critical rows are the rows 1, 3 and 6. After removing S_1, the critical rows are the rows 1, 3 and 6 again, where the row-complexity is 2 now. The next two steps of the segmentation procedure then

might produce

$$S_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } S_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

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

Obviously it remains the question whether **MIN-DT** for $\mathcal{S}' = \mathcal{S}_{TG}$ is still polynomial for integer input matrices. Up to now, we do not see that the tools we developed here can be generalized to provide a result for the integer case.

4.3.2 Relation to colorings of perfect graphs

We show that finding the optimal delivery time of a TG-segmentation of a binary matrix is equivalent to computing the chromatic number in a perfect graph. This yields an alternative proof that the problem can be solved in polynomial time.
The *chromatic number* of a graph $G = (V, E)$ is the minimal number of colors we need to color the vertices of G such that no two adjacent vertices have the same color. This number is denoted by $\chi(G)$. A *clique* in G is a subset of vertices, such that each two of them are adjacent. The size of a largest clique in G is denoted by $\omega(G)$. A *stable set* in G is a set of vertices such that each two of them are not adjacent.
A *perfect graph* is a graph G in which the chromatic number of every induced subgraph equals the size of the largest clique of that subgraph, i.e. G is perfect if for every induced subgraph G' of G we have $\chi(G') = \omega(G')$.
Let the binary matrix A be given. We define a graph $G_A = (V_A, E_A)$ as follows: The set of vertices V_A is the set of boxes of A. The set of edges E_A is the set of pairs of boxes (B, B') such that B and B' are not allowed to be in the same segment of a TG-segmentation of A. That means, two boxes $B = [i_1, i_2] \times \{j\}$ and $B' = [i'_1, i'_2] \times \{j'\}$ are adjacent, if $[i_1, i_2] \cap [i'_1, i'_2] \neq \emptyset$ and if either

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

For example, in Figure 4.10 box 4 and 10 from the left would be adjacent, because the boxes 7 to 10 form an i-cap. Note that if two boxes belong to the same connected region of ones resulting from the splitting procedure, they are not adjacent in the graph. This graph is called the *TG-graph* of A.

Theorem 5. $c^{TG}(A) = \omega(G_A)$

Proof. $c^{TG}(A) \geq \omega(G_A)$ is easy to see, as each box in a maximal clique of G_A needs its own segment. Let now i be a row with $c_i^{TG}(A) = c^{TG}(A)$, i.e. after applying the splitting procedure the boxes of row i decompose into $c^{TG}(A)$ many connected regions of ones such that each two of them have to be irradiated separately. As we show now, it is possible to choose $c^{TG}(A)$ boxes from row i that form a clique in G_A. The choice of the clique is illustrated in Figure 4.10 and can be realized as follows:

- We go through the boxes $B_{i1}, \ldots, B_{in}$ of row i from left to right and insert $s_i(A)$ splits such that the splits occur as late as possible, i.e. we insert a split whenever there would be a connected region of ones that does not satisfy the consecutive ones property otherwise.
- W.l.o.g. we only have to discuss the choice of boxes within a sequence $B_{ij}, \ldots, B_{ij'}$ of nonempty boxes (i.e. $a_{ik} \neq 0$ for $j \leq k \leq j'$), because if there is a zero in row i between two boxes they are adjacent in the graph anyway. Let us therefore consider such a sequence of consecutive connected nonempty boxes.
- In each such sequence, we pick the first box of each connected region of ones, i.e. the very first box of the sequence and every box that comes directly after a split.
- This indeed gives a clique, because if a chosen box could be in a common segment with the previously chosen one, we could move the split one column to the right, a contradiction to the choice of the splits. As each chosen box cannot be in the same segment with the previously chosen one, it of course also cannot be in the same segment with all other boxes chosen before.

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

□

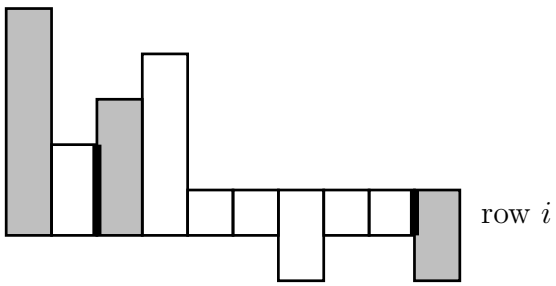

Fig. 4.10: Choice of a maximal clique. The bold lines indicate the split positions and the grey boxes are chosen to form a maximal clique.

Theorem 6. $c^{TG}(A) = \chi(G_A)$

Proof. By definition, the chromatic number is the minimal number of stable sets we need to decompose a graph, as each color has to be assigned to a stable set of vertices. Obviously, $\chi(G_A) \leq c^{TG}(A)$, as each segment exactly corresponds to a stable set in G_A and therefore an optimal segmentation yields a coloring with $c^{TG}(A)$ many colors. Furthermore, we have $\omega(G_A) \leq \chi(G_A)$, as this holds for every graph. Together with Theorem 5, we have

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

and thus $c^{TG}(A) = \chi(G_A)$. □

Theorems 5 and 6 together give $\chi(G_A) = \omega(G_A)$ for the TG-graph G_A of A. If we consider induced subgraphs of G_A, the boxes that correspond to the chosen subset of vertices form a binary matrix, that we call A' from now on. Note that the induced subgraph of G_A, denoted by H, that has the boxes of A' as vertices, is not necessarily $G_{A'}$. It can happen, that two boxes B and B' of A' are adjacent in $G_{A'}$, but not adjacent in the induced subgraph H, because other boxes of A, that do not belong to A', allowed putting B and B' into the same segment. This is illustrated in Figure 4.11.

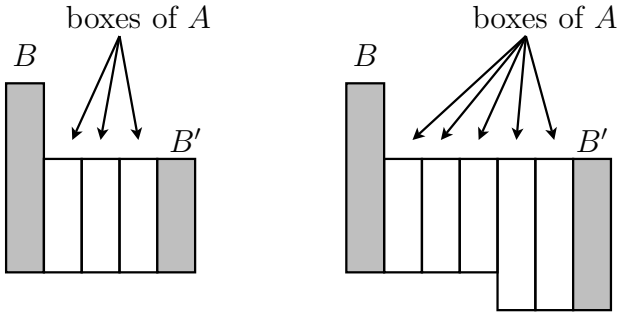

Fig. 4.11: Example for boxes of A and A'. The white boxes are boxes of A that are substituted by zero entries in A'. These are two examples where B and B' are not adjacent in the induced subgraph, because there were other boxes of A that made their combination possible.

Theorem 7. *For every induced subgraph H of G_A there exists a graph $G_{A''}$ that is the TG-graph of a binary matrix A'' with $\chi(H) = \chi(G_{A''})$ and $\omega(H) = \omega(G_{A''})$.*

Proof. Let H be an arbitrary induced subgraph of G_A such that the boxes of the induced vertex set form a binary matrix A'. Let A'' be the matrix that has the same boxes as A' and some extra boxes defined as follows: Whenever we have two boxes $B = [i_1, i_2] \times \{j\}$ and $B' = [k_1, k_2] \times \{\ell\}$ in A' with $\ell > j$ such that B and B' are not adjacent in G_A and such that there are only zeros in $([i_1, i_2] \cap [k_1, k_2]) \times [j+1, \ell-1]$ in A', then we add the intermediate boxes $([i_1, i_2] \cap [k_1, k_2]) \times \{t\}$ for all $t \in [j+1, \ell-1]$ to A''. For example, the three white boxes in Figure 4.11 on the left are the intermediate boxes of B and B'. Therefore, B and B' can be put into the same segment of a TG-segmentation of A''. Let $G_{A''}$ be the TG-graph corresponding to the matrix A''. $G_{A''}$

has more vertices than H and some extra edges that are incident with the new vertices. Note that two boxes of A' that are adjacent in H are also adjacent in $G_{A''}$, because they still cannot be in the same segment. Similarly, two boxes of A' that are not adjacent in H are not adjacent in $G_{A''}$, as we inserted the intermediate boxes. Thus, H is an induced subgraph of $G_{A''}$ and $\chi(H) \leq \chi(G_{A''})$ as well as $\omega(H) \leq \omega(G_{A''})$ is immediately obvious.
Let now B and B' be two such boxes of A' such that we inserted the intermediate boxes between them in A''. It is easy to verify that the intermediate boxes are not adjacent to B and B' in $G_{A''}$ and also not adjacent to all non-neighbors of B and B' in $G_{A''}$ (and also in H, as these non-neighbors are the same). This is the case, because the intermediate boxes can be put into the same segment with all boxes that can be put into the same segment with either B or B'. Thus, if we have an optimal decomposition of H into stable sets, we can put all the intermediate boxes into the stable set containing B or the stable set containing B' and get stable sets in $G_{A''}$. Doing this for all pairs B and B' where we have intermediate boxes yields a stable set decomposition of $G_{A''}$ with $\chi(H)$ many stable sets. Thus, $\chi(H) \geq \chi(G_{A''})$.
Let us now consider a largest clique in $G_{A''}$. If this clique contains no intermediate boxes, this is a clique in H. If it contains intermediate boxes, we do the following substitution: For every boxes B and B' of A' where we have intermediate boxes in between, only either B or B' or one of the intermediate boxes can be in the clique, as they are all not adjacent. If an intermediate box is contained in the maximal clique, we delete the intermediate box and put either B or B' into the maximal clique. This is possible, because every box that cannot be in the same segment with the intermediate box also cannot be in the same segment with B and B' and thus all neighbors of the intermediate box are also neighbors of B and B'. The question arises if we might need a box B twice for substitution because there are intermediate boxes left and right from B. But this cannot happen because in a sequence B'',B, B' with intermediate boxes between B'' and B and between B and B', all intermediate boxes (left and right from B) and B are not adjacent (as they can be in the same segment). Therefore, there never can be two intermediate boxes of the sequence B'', B, B' with intermediate boxes between B'' and B and between B and B' in a maximal clique. After the substitution procedure we have found a clique of the same cardinality containing only boxes from A'. These boxes form a clique in H and thus we get $\omega(H) \geq \omega(G_{A''})$. This concludes the proof. □

Using Theorems 5, 6 and 7 we get the following

Corollary 8. *The graph G_A is a perfect graph with $\chi(G_A) = c^{TG}(A)$.*

As the coloring problem in perfect graphs can be solved in polynomial time, the delivery time problem for TG-segmentations is also polynomial. We remark that, although the chromatic number of G_A gives the optimal delivery time of a TG-segmentation of A, not all optimal colorings of G_A yield a TG-segmentation of A. For example, if the stable set decomposition is like in Figure 4.12 on the left, we get no feasible TG-segmentation, as there are two boxes in a stable set that only form a segment if the intermediate boxes are in the same stable set (like in the decomposition on the right).

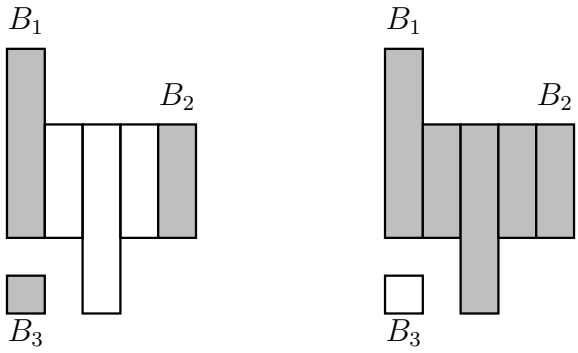

Fig. 4.12: The same boxes with two different stable set decompositions (the grey boxes and the white boxes each form a stable set). The decomposition on the right gives a TG-segmentation, but the decomposition on the left does not.

Thus, we should ask ourselves how we can modify an optimal stable set decomposition of G_A in such a way that each stable set really represents a segment. But from an algorithmic point of view, this question is not interesting, as the algorithms that solve the coloring problem in a perfect graph are slower than the one we have presented in Section 4.3.1.

5. APPROXIMATE DISCRETE SEGMENTATION FOR DELIVERY TIME MINIMIZATION

In clinical practice, if the delivery time of an exact decomposition is too large, one might admit certain deviations between the target matrix and the realized matrix. These deviations certainly depend on the position in the matrix, because different allowed dose deviations are necessary for the target volume, the organs at risk and the rest of the body. Furthermore, an underdosage for the target volume might not be acceptable, while a small overdosage can be permitted. In this section, we solve the problems **Approx-MIN-DT** and **Approx-MIN-DT-TC**, i.e. we have given a lower bound matrix $\underline{A}$ and an upper bound matrix $\overline{A}$ and require

$$\underline{a}_{i,j} \leq b_{i,j} \leq \overline{a}_{i,j}$$

for all $(i,j) \in [m] \times [n]$ for the approximation matrix B. The aim is to minimize the delivery time of the segmentation of B for **Approx-MIN-DT** and to find an approximation realizing this minimum delivery time with minimal total change for **Approx-MIN-DT-TC**. We solely deal with the ℓ_1-norm as measure for the total change. In Section 5.1 we solve the unconstrained problem where all segments are feasible and the DT of the approximation matrix is just $c(B)$ which was defined in (4.1). Section 5.2 deals with the case that $\mathcal{S}' = \mathcal{S}_{ICC}$.

5.1 Delivery time minimization in general

The results of this section are a generalization of the approaches in [26] where the allowed deviations were equal for each (i,j) and for under- and overdosage, i.e. we just required $(a_{ij} - \delta)_+ \leq b_{ij} \leq a_{ij} + \delta$ for some given $\delta \in \mathbb{N}$. Note that our problems can be rewritten using the notation with the deviations from [26] by setting

$$\underline{\delta}_{i,j} = a_{i,j} - \underline{a}_{i,j} \quad \text{and} \quad \overline{\delta}_{i,j} = \overline{a}_{i,j} - a_{i,j}.$$

The resulting problem is then, given A, $\underline{\Delta} = (\underline{\delta}_{i,j})$ and $\overline{\Delta} = (\overline{\delta}_{i,j})$, to find an approximation B with $a_{i,j} - \underline{\delta}_{i,j} \leq b_{i,j} \leq a_{i,j} + \overline{\delta}_{i,j}$. Let the corresponding problems be called **Approx-MIN-DT-Deltas** and **Approx-MIN-DT-TC-Deltas**. Note that our notation is indeed a bit more general, as the solution of **Approx-MIN-DT** does not at all depend on the matrix A. Several instances of **Approx-MIN-DT-Deltas** are translated to the same instance of **Approx-MIN-DT**, because A can be shifted between $\underline{A}$ and $\overline{A}$. The problems **Approx-MIN-DT-TC** and **Approx-MIN-DT-TC-Deltas** are more or less the same. Nevertheless, it is easy to see, that also **Approx-MIN-DT**

and **Approx-MIN-DT-Deltas** are equivalent, as if one can solve one of them, one can also solve the other one by an easy transformation.
For a vector $\boldsymbol{a} \in \mathbb{Z}_+^n$, let $c(\boldsymbol{a}) = \sum_{j=1}^{n}(a_j - a_{j-1})_+$, where $a_0 = 0$. We neglect further technical constraints for the moment and can therefore model the rows of A independently. The derived single row problems are then the following.

Approx-MIN-DT-Row: Given the vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$, $\underline{\boldsymbol{a}} = (\underline{a}_1, \ldots, \underline{a}_n)$ and $\overline{\boldsymbol{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ with nonnegative integral entries and

$$\underline{a}_j \leq a_j \leq \overline{a}_j \quad \forall\ j \in [n],$$

find a vector $\boldsymbol{b}$ with nonnegative integral entries such that $\underline{a}_j \leq b_j \leq \overline{a}_j$ and $c(\boldsymbol{b})$ is minimum.

Let $c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(\boldsymbol{a})$ denote this minimum.

Approx-MIN-DT-TC-Row: Given the vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$, $\underline{\boldsymbol{a}} = (\underline{a}_1, \ldots, \underline{a}_n)$ and $\overline{\boldsymbol{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ with nonnegative integral entries and

$$\underline{a}_j \leq a_j \leq \overline{a}_j \quad \forall\ j \in [n],$$

find a vector $\boldsymbol{b}$ with nonnegative integral entries such that $\underline{a}_j \leq b_j \leq \overline{a}_j$, $c(\boldsymbol{b}) = c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(\boldsymbol{a})$ and $\|\boldsymbol{a} - \boldsymbol{b}\|_1$ is minimum.

5.1.1 Solution of Approx-MIN-DT-Row

In the following all intervals are subsets of $[0, n+1]$. Let appropriate vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$, $\underline{\boldsymbol{a}} = (\underline{a}_1, \ldots, \underline{a}_n)$ and $\overline{\boldsymbol{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ be given and let always $a_0 = \underline{a}_0 = \overline{a}_0 = a_{n+1} = \underline{a}_{n+1} = \overline{a}_{n+1} = 0$.
In the forthcoming definitions, we will leave out indices that show us that the quantities depend on $\underline{\boldsymbol{a}}$ and $\overline{\boldsymbol{a}}$. Assume that the vectors are fixed and always remind yourself that all quantities are computed with regard to them.
For any $j \in [0, n+1]$ we define its *lower level interval* $\underline{I}(j)$ and its *upper level interval* $\overline{I}(j)$ to be the maximal interval containing j such that

$$\underline{a}_i \leq \overline{a}_j \quad \forall i \in \underline{I}(j),$$
$$\overline{a}_i \geq \underline{a}_j \quad \forall i \in \overline{I}(j),$$

respectively. An element $j \in [0, n+1]$ is called a *local minimum for* $\boldsymbol{a}$ if

$$\overline{a}_j \leq \overline{a}_i \quad \forall i \in \underline{I}(j),$$

and it is called a *local maximum for* $\boldsymbol{a}$ if

$$\underline{a}_j \geq \underline{a}_i \quad \forall i \in \overline{I}(j).$$

We say that j is a *local extremum* if it is a local minimum or a local maximum.
For our further investigations, we need some lemmas about the structure of the minima and maxima.

Lemma 6. *If $\underline{\boldsymbol{a}} \neq \mathbf{0}$, no element $j \in [0, n+1]$ can simultaneously be a local minimum and a local maximum.*

Proof. Let $\underline{\boldsymbol{a}} \neq \mathbf{0}$ and let j be a local minimum. We consider the intervals $\underline{I}(j) = [i_1, k_1]$ and $\overline{I}(j) = [i_2, k_2]$ containing j. Let $[i,k] = [i_1, k_1] \cap [i_2, k_2]$. Assume j is also a local maximum. Thus, $\overline{a}_\ell \geq \overline{a}_j$ and $\underline{a}_\ell \leq \underline{a}_j$ for all $\ell \in [i,k]$.
Case 1. $[i,k] = [0, n+1]$. Thus, $0 = \overline{a}_0 \geq \overline{a}_j \geq \underline{a}_j$ and therefore $\overline{a}_j = \underline{a}_j = 0$. This yields $\underline{a}_\ell = 0$ for all $\ell \in [0, n+1]$, which is a contradiction to $\underline{\boldsymbol{a}} \neq \mathbf{0}$.
Case 2. $i > 0$ or $k < n+1$. W.l.o.g. let $i > 0$. Either $\underline{a}_{i-1} > \overline{a}_j$ (*) (if $i - 1 \notin \underline{I}(j)$) or $\overline{a}_{i-1} < \underline{a}_j$ (**) (if $i - 1 \notin \overline{I}(j)$). In the case of (*), we get $i - 1 \in \overline{I}(j)$ and $\overline{a}_{i-1} \geq \underline{a}_{i-1} > \overline{a}_j$, a contradiction to j being a local maximum. In the case of (**), we get $i - 1 \in \underline{I}(j)$ and $\underline{a}_{i-1} \leq \overline{a}_{i-1} < \underline{a}_j$, contradicting j being a local minimum. □

Note that if $\underline{\boldsymbol{a}} = \mathbf{0}$, then the optimal approximation vector both for **Approx-MIN-DT-Row** and **Approx-MIN-DT-TC-Row** is the zero vector and the problem is easy. Therefore, the condition $\underline{\boldsymbol{a}} \neq \mathbf{0}$ in the previous lemma is not a restriction.

We say that two local extrema i and j where $0 \leq i < j \leq n+1$ are *consecutive* if there is no $k \in [i+1, j-1]$ that is a local extremum. Note that consecutive does *not* mean that $j = i + 1$.

Lemma 7. *Let $0 \leq i < j \leq n+1$ and let i, j be consecutive local extrema.*

a) If i and j are local minima then $\overline{a}_i = \overline{a}_j$ and $\underline{I}(i) = \underline{I}(j) \supseteq [i,j]$.

b) If i and j are local maxima then $\underline{a}_i = \underline{a}_j$ and $\overline{I}(i) = \overline{I}(j) \supseteq [i,j]$.

Proof. We prove only a) because b) can be treated analogously.
Case 1. $\underline{a}_k \leq \overline{a}_i$ for all $k \in [i,j]$. Then $j \in \underline{I}(i)$ and hence $\overline{a}_j \geq \overline{a}_i$. But then also $\underline{a}_k \leq \overline{a}_i \leq \overline{a}_j$ for all $k \in [i,j]$ and consequently $i \in \underline{I}(j)$, which implies $\overline{a}_i \geq \overline{a}_j$. Accordingly, $\overline{a}_i = \overline{a}_j$ and since $[i,j] \subseteq \underline{I}(i) \cap \underline{I}(j)$, we immediately obtain $\underline{I}(i) = \underline{I}(j)$.
Case 2. There exists a $k \in [i,j]$ such that $\underline{a}_k > \overline{a}_i$. We may assume that $\underline{a}_k = \max\{\underline{a}_\ell \ : \ \ell \in [i,j]\}$.
Case 2.1. $\underline{a}_k > \overline{a}_j$. Then $\overline{a}_i < \underline{a}_k$ and $\overline{a}_j < \underline{a}_k$ which implies that $\overline{I}(k) \subseteq [i+1, j-1]$ and hence k is a local maximum between i and j, a contradiction to i and j being consecutive local extrema.
Case 2.2. $\underline{a}_k \leq \overline{a}_j$. Then $\ell \in \underline{I}(j)$ for all $\ell \in [i,j]$ and in particular $i \in \underline{I}(j)$. This implies $\overline{a}_i \geq \overline{a}_j$ and $\underline{a}_k \leq \overline{a}_j \leq \overline{a}_i$, a contradiction to the condition of Case 2. □

Case 2.1 from the proof can be generalized in the following way:

Lemma 8. *Let $0 \leq i < k < j \leq n+1$.*

a) If $\underline{a}_k > \overline{a}_i$ and $\underline{a}_k > \overline{a}_j$ then there is a local maximum between i and j.

b) If $\overline{a}_k < \underline{a}_i$ and $\overline{a}_k < \underline{a}_j$ then there is a local minimum between i and j.

Proof. We prove only a) because b) can be treated analogously. Choose $h \in [i,j]$ such that $\underline{a}_h = \max\{\underline{a}_\ell \ : \ \ell \in [i,j]\}$. Then, by supposition, also $\underline{a}_h > \overline{a}_i$ and $\underline{a}_h > \overline{a}_j$. Hence $\overline{I}(h) \subseteq [i+1, j-1]$ and h is a local maximum between i and j. □

Lemma 9. *Let* $\min\{\overline{a}_k \ : \ k \in [n]\} < \max\{\underline{a}_k \ : \ k \in [n]\}$. *Let* $0 \leq i < j \leq n+1$ *and let* i, j *be consecutive local extrema.*

a) If i *is a local minimum and* j *is a local maximum then* $\overline{a}_i \leq \overline{a}_\ell$ *and* $\underline{a}_\ell \leq \underline{a}_j$ *for all* $\ell \in [i,j]$ *and* $\underline{a}_j > \overline{a}_i$.

b) If i *is a local maximum and* j *is a local minimum then* $\underline{a}_i \geq \underline{a}_\ell$ *and* $\overline{a}_\ell \geq \overline{a}_j$ *for all* $\ell \in [i,j]$ *and* $\underline{a}_i > \overline{a}_j$.

Proof. We prove only a) because b) can be treated analogously.
1. Let $k \in [i,j]$ such that $\overline{a}_k = \min\{\overline{a}_\ell \ : \ \ell \in [i,j]\}$. Assume that $\overline{a}_k < \overline{a}_i$. (The case of a maximum and $\underline{a}_k > \underline{a}_j$ is analogous).
Case 1. $\underline{a}_h \leq \overline{a}_k$ for all $h \in [i,k]$. Then also $\underline{a}_h \leq \overline{a}_i$ for all $h \in [i,k]$ and hence $k \in \underline{I}(i)$. But then $\overline{a}_k < \overline{a}_i$ contradicts the fact that i is a local minimum.
Case 2. There is some $h \in [i,k]$ with $\underline{a}_h > \overline{a}_k$.
Case 2.1. $\underline{a}_j > \overline{a}_k$. Then, by Lemma 8 b), there is a local minimum between h and j and hence also between i and j. This is a contradiction to i and j being consecutive local extrema.
Case 2.2. $\underline{a}_j \leq \overline{a}_k$. Then $\overline{a}_\ell \geq \underline{a}_j$ for all $\ell \in [i,j]$ and in particular $h \in \overline{I}(j)$. But $\underline{a}_h > \overline{a}_k \geq \underline{a}_j$ is a contradiction since j is a local maximum.
2. Now assume that $\underline{a}_j \leq \overline{a}_i$. Since $\min\{\overline{a}_k \ : \ k \in [n]\} < \max\{\underline{a}_k \ : \ k \in [n]\}$ there is an element $k \in [0, n+1]$ such that $\overline{a}_k < \overline{a}_i$ or $\underline{a}_k > \underline{a}_j$. We already know that necessarily $k \notin [i,j]$. We may assume that k is the nearest element to the interval $[i,j]$ with the property $\overline{a}_k < \overline{a}_i$ or $\underline{a}_k > \underline{a}_j$. W.l.o.g. let $k < i$. The case $k > j$ is analogous.
Case 1. $\overline{a}_k < \overline{a}_i$. Then, by the choice of k, $\underline{a}_\ell \leq \underline{a}_j$ for all $\ell \in [k+1, i]$ and hence, by the assumption $\underline{a}_j \leq \overline{a}_i$, we have $\underline{a}_\ell \leq \overline{a}_i$ for all $\ell \in [k,i]$. Accordingly, $k \in \underline{I}(i)$ and $\overline{a}_k < \overline{a}_i$ contradicts to i being a local minimum.
Case 2. $\underline{a}_k > \underline{a}_j$ and Case 1 does not hold. Then, by the choice of k, $\overline{a}_\ell \geq \overline{a}_i$ for all $\ell \in [k+1, i]$ and by supposition of this case and 1. moreover for all $\ell \in [k,j]$. Hence, by the assumption $\underline{a}_j \leq \overline{a}_i$, we have $\overline{a}_\ell \geq \underline{a}_j$ for all $\ell \in [k,j]$. Accordingly, $k \in \overline{I}(j)$ and $\underline{a}_k > \underline{a}_j$ contradicts to j being a local maximum. □

We already know that consecutive local extrema i and j either have different types or, if both are minima (respectively maxima), the corresponding upper bounds $\overline{a}_i$ and $\overline{a}_j$ (respectively lower bounds $\underline{a}_i$ and $\underline{a}_j$) are equal (using Lemma 7) and $\underline{a}_k \leq \overline{a}_i = \overline{a}_j$ (respectively $\overline{a}_k \geq \underline{a}_i = \underline{a}_j$) for all $k \in [i,j]$ (using Lemma 8). This shows us, that it is possible to make the sequence of entries between two consecutive minima (respectively maxima) constant.
For each not extendable sequence of consecutive local minima and for each not extendable sequence of consecutive local maxima we pick the first and the last one. In such a way we obtain an alternating sequence

$$\boldsymbol{s} = (0 = m_1, m^1, M_1, M^1, m_2, m^2, M_2, M^2, \ldots, M_t, M^t, m_{t+1}, m^{t+1} = n+1)$$

of pairs of local minima and maxima. We call $\boldsymbol{s}$ the *min-max sequence* of $\boldsymbol{a}$. We note that $m_\ell = m^\ell$ and $M_\ell = M^\ell$ is allowed - this is the case if the corresponding sequence

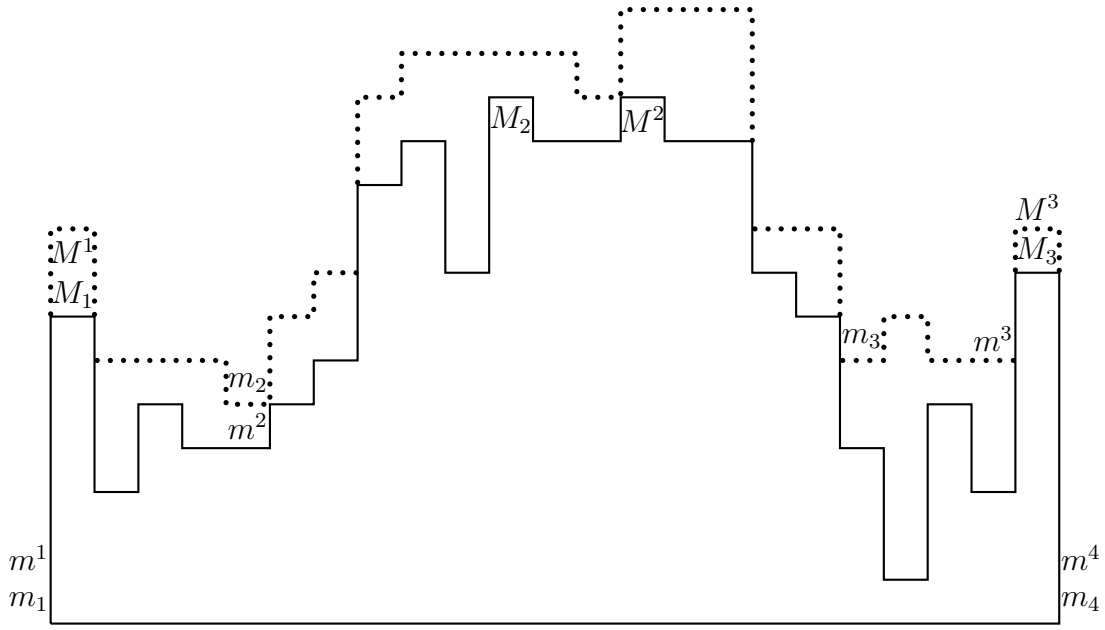

Fig. 5.1: The min-max sequence of a vector. The vector with the solid line is $\underline{\overline{\boldsymbol{a}}}$ and the dotted line represents $\overline{\boldsymbol{a}}$.

of consecutive local extrema contains only a single element. Moreover, as an exception, we set $m^1 = m_1 = 0$ and $m_{t+1} = m^{t+1} = n + 1$.
Algorithm 9 from the appendix computes the min-max sequence of $\boldsymbol{a}$ in $O(n)$ time. From Lemma 7 we immediately obtain:

Lemma 10.

a) For all $\ell \in [1, t+1]$, we have $\overline{a}_{m_\ell} = \overline{a}_{m^\ell}$ and $\underline{I}(m_\ell) = \underline{I}(m^\ell) \supseteq [m_\ell, m^\ell]$.

b) For all $\ell \in [1, t]$, we have $\underline{a}_{M_\ell} = \underline{a}_{M^\ell}$ and $\overline{I}(M_\ell) = \overline{I}(M^\ell) \supseteq [M_\ell, M^\ell]$.

We say that a vector $\boldsymbol{b}$ is *conform* to the min-max sequence if

$$\begin{aligned}
&b_j = \overline{a}_{m_\ell} \text{ for all } j \in [m_\ell, m^\ell], \quad \ell = 2, \ldots, t,\\
&b_j = \underline{a}_{M_\ell} \text{ for all } j \in [M_\ell, M^\ell], \quad \ell = 1, \ldots, t,\\
&b_j \geq b_{j-1} \text{ for all } j \in [m^\ell + 1, M_\ell], \quad \ell = 1, \ldots, t,\\
&b_j \geq b_{j+1} \text{ for all } j \in [M^\ell, m_{\ell+1} - 1], \quad \ell = 1, \ldots, t.
\end{aligned}$$

Hence, if $\boldsymbol{b}$ is conform to the min-max sequence, then $\boldsymbol{b}$ is constant in the intervals $[m_\ell, m^\ell], [M_\ell, M^\ell]$, increasing in the intervals $[m^\ell, M_\ell]$ and decreasing in the intervals $[M^\ell, m_{\ell+1}]$.

Lemma 11. *Let $\min\{\overline{a}_k : k \in [n]\} < \max\{\underline{a}_k : k \in [n]\}$ and let $\boldsymbol{b}$ be a feasible solution of the problem* **Approx-MIN-DT-Row**. *Then*

$$c(\boldsymbol{b}) \geq \underline{a}_{M_1} + \sum_{\ell=2}^{t} (\underline{a}_{M_\ell} - \overline{a}_{m_\ell}).$$

If equality holds then $\boldsymbol{b}$ is conform to the min-max sequence.

Proof. We have

$$\begin{aligned} b_{M_\ell} &\geq \underline{a}_{M_\ell}, \quad \ell = 1, \ldots, t, \\ b_{m_\ell} &\leq \overline{a}_{m_\ell}, \quad \ell = 2, \ldots, t \end{aligned}$$

and hence

$$\begin{aligned} c(\boldsymbol{b}) &= \sum_{j=1}^{n} (b_j - b_{j-1})_+ && (5.1) \\ &\geq \sum_{j=1}^{M_1} (b_j - b_{j-1})_+ + \sum_{\ell=2}^{t} \sum_{j=m_\ell+1}^{M_\ell} (b_j - b_{j-1})_+ && (5.2) \\ &\geq \sum_{j=1}^{M_1} (b_j - b_{j-1}) + \sum_{\ell=2}^{t} \sum_{j=m_\ell+1}^{M_\ell} (b_j - b_{j-1}) && (5.3) \\ &= (b_{M_1} - 0) + \sum_{\ell=2}^{t} (b_{M_\ell} - b_{m_\ell}) && (5.4) \\ &\geq \underline{a}_{M_1} + \sum_{\ell=2}^{t} (\underline{a}_{M_\ell} - \overline{a}_{m_\ell}). && (5.5) \end{aligned}$$

If not $b_{m_\ell} = b_{m^\ell} = \overline{a}_{m_\ell}$ and $b_{M_\ell} = b_{M^\ell} = \underline{a}_{M_\ell}$ for all ℓ, then inequality (5.5) is strict. Otherwise, if these values are correct, but $\boldsymbol{b}$ is not constant in an interval $[m_\ell, m^\ell]$ or $[M_\ell, M^\ell]$ then $b_j - b_{j-1} > 0$ for some $j \in [m_\ell + 1, m^\ell]$ or $j \in [M_\ell + 1, M^\ell]$ and thus inequality (5.2) is strict. If $\boldsymbol{b}$ is not decreasing in an interval $[M^\ell, m_{\ell+1}]$ then $b_j - b_{j-1} > 0$ for some $j \in [M^\ell + 1, m_{\ell+1}]$ and thus again inequality (5.2) is strict. If $\boldsymbol{b}$ is not increasing in an interval $[m^\ell, M_\ell]$ then $b_j - b_{j-1} < 0$ for some $j \in [m^\ell + 1, M_\ell]$ and inequality (5.3) is strict. □

We construct two vectors $\underline{\boldsymbol{b}}$ and $\overline{\boldsymbol{b}}$ which turn out to be conform to the min-max sequence as follows: We set

$$\begin{aligned} \underline{b}_0 = \overline{b}_0 &= 0, \\ \underline{b}_{n+1} = \overline{b}_{n+1} &= 0, \\ \underline{b}_j = \overline{b}_j &= \overline{a}_{m_\ell}, \quad j \in [m_\ell, m^\ell], \quad \ell = 2, \ldots, t, \\ \underline{b}_j = \overline{b}_j &= \underline{a}_{M_\ell}, \quad j \in [M_\ell, M^\ell], \quad \ell = 1, \ldots, t. \end{aligned}$$

Using this initialization, we set

$$\underline{b}_j = \max\{\underline{b}_{j-1}, \underline{a}_j\}, \quad j = m^\ell + 1, m^\ell + 2, \ldots, M_\ell - 1, \quad \ell = 1, \ldots, t, \tag{5.6}$$

$$\underline{b}_j = \max\{\underline{b}_{j+1}, \underline{a}_j\}, \quad j = m_{\ell+1} - 1, m_{\ell+1} - 2, \ldots, M^\ell + 1, \quad \ell = 1, \ldots, t. \tag{5.7}$$

and

$$\overline{b}_j = \min\{\overline{b}_{j+1}, \overline{a}_j\}, \quad j = M_\ell - 1, M_\ell - 2, \ldots, m^\ell + 1, \quad \ell = 1, \ldots, t, \tag{5.8}$$

$$\overline{b}_j = \min\{\overline{b}_{j-1}, \overline{a}_j\}, \quad j = M^\ell + 1, M^\ell + 2, \ldots, m_{\ell+1} - 1, \quad \ell = 1, \ldots, t. \tag{5.9}$$

Note that in (5.6) and (5.9) the iteration is computed forwards, and in (5.7) and (5.8) backwards. This is indeed possible, as we initialized all starting values of the iterations.

Example 5. We use the vectors corresponding to Figure 5.1. For

$$\begin{aligned}\underline{\boldsymbol{a}} &= (7,3,5,4,4,5,6,10,11,8,12,11,11,12,11,11,8,7,4,1,5,3,8)\\ \overline{\boldsymbol{a}} &= (9,6,6,6,5,7,8,12,13,13,13,13,12,14,14,14,9,9,6,7,6,6,9)\end{aligned}$$

the sequence $\boldsymbol{s}$ has the form

$$\boldsymbol{s} = (0,0,1,1,5,5,11,14,19,22,23,23,24,24)$$

and we get

$$\begin{aligned}\underline{\boldsymbol{b}} &= (7|5,5,5|5|5,6,10,11,11|12,12,12,12|11,11,8,7|6,6,6,6|8)\\ \overline{\boldsymbol{b}} &= (7|6,6,6|5|7,8,12,12,12|12,12,12,12|12,12,9,9|6,6,6,6|8).\end{aligned}$$

The vertical bars represent the regions $[m_\ell, m^\ell]$ and $[M_\ell, M^\ell]$. Figure 5.2 illustrates the vectors $\underline{\boldsymbol{b}}$ and $\overline{\boldsymbol{b}}$.

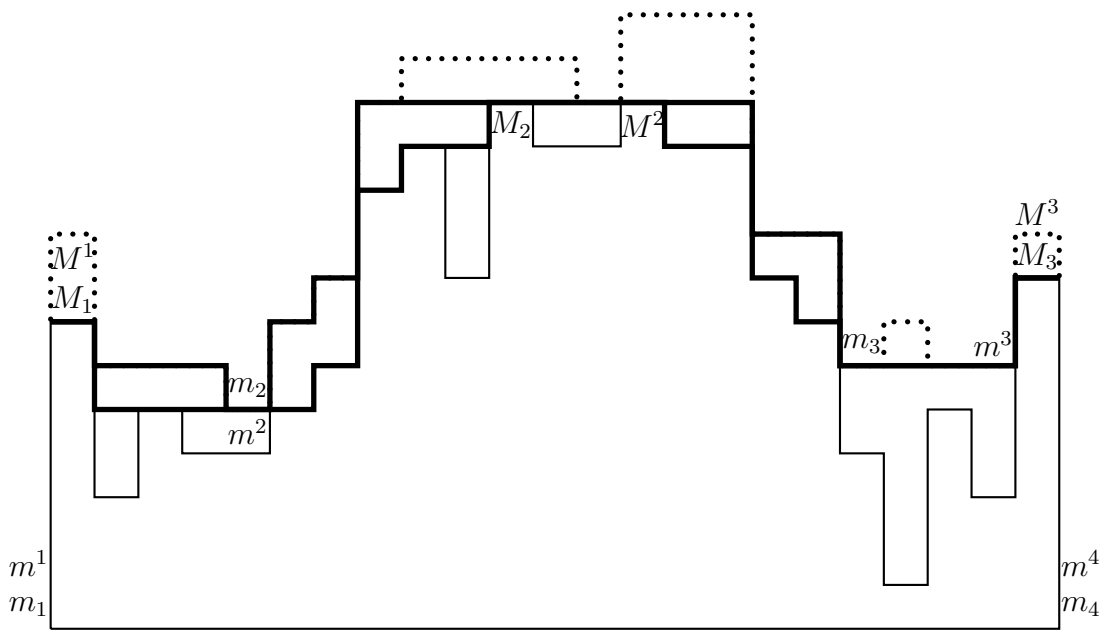

Fig. 5.2: An example for the min-max sequence with the corresponding extremal optimal vectors $\underline{\boldsymbol{b}}$ and $\overline{\boldsymbol{b}}$ (printed with thick lines).

Theorem 9. *Let* $\min\{\overline{a}_k \;:\; k \in [n]\} < \max\{\underline{a}_k \;:\; k \in [n]\}$. *The vectors* $\underline{\boldsymbol{b}}$ *and* $\overline{\boldsymbol{b}}$ *are optimal solutions for the problem* **Approx-MIN-DT-Row**. *An arbitrary vector* $\boldsymbol{b}$ *is an optimal solution for the problem* **Approx-MIN-DT-Row** *iff* $\boldsymbol{b}$ *is conform to the min-max sequence and* $\underline{\boldsymbol{b}} \le \boldsymbol{b} \le \overline{\boldsymbol{b}}$. *We have*

$$c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(\boldsymbol{a}) = \underline{a}_{M_1} + \sum_{\ell=2}^{t} (\underline{a}_{M_\ell} - \overline{a}_{m_\ell}).$$

Proof. 1. First we show that $\underline{\boldsymbol{b}}$ is feasible. We have to prove that $\underline{a}_j \leq \underline{b}_j \leq \overline{a}_j$ for all $j \in [n]$.
Case 1. $j \in [m_\ell, m^\ell]$ for some $\ell \in [2,t]$. By Lemma 10, $\underline{a}_j \leq \overline{a}_{m_\ell} \leq \overline{a}_j$ and consequently $\underline{a}_j \leq \overline{a}_{m_\ell} = \underline{b}_j = \overline{b}_j \leq \overline{a}_j$.
Case 2. $j \in [M_\ell, M^\ell]$ for some $\ell \in [1,t]$. This case is analogous to Case 1.
Case 3. $j \in [m^\ell + 1, M_\ell - 1]$ for some $\ell \in [1,t]$. By construction, we have $\underline{a}_j \leq \underline{b}_j$. Assume that $\underline{b}_j > \overline{a}_j$. Let i be the smallest index from $[m^\ell, j]$ such that $\underline{b}_i = \underline{b}_{i+1} = \cdots = \underline{b}_j$. Then $\underline{b}_i > \overline{a}_j$.
Case 3.1. $i = m^\ell$. Then $\underline{b}_{m^\ell} > \overline{a}_j$. If $\ell = 1$ we immediately get the contradiction $0 > \overline{a}_j$. If $\ell > 1$, we have $\underline{b}_{m^\ell} = \overline{a}_{m^\ell} > \overline{a}_j$, a contradiction to Lemma 9 a).
Case 3.2. $i > m^\ell$. Then $\underline{b}_{i-1} < \underline{b}_i = \underline{a}_i > \overline{a}_j$. By Lemma 9 a), $\underline{a}_i \leq \underline{a}_{M_\ell}$ and consequently also $\underline{a}_{M_\ell} > \overline{a}_j$. From Lemma 8 we obtain that there is a local minimum between i and M_ℓ and hence also between m^ℓ and M_ℓ, a contradiction.
Case 4. $j \in [M^\ell + 1, m_{\ell+1} - 1]$ for some $\ell \in [1,t]$. This case is analogous to Case 3.

2. Now we show that $\underline{\boldsymbol{b}}$ is conform to the min-max sequence. The construction implies that it is sufficient to verify that $\underline{b}_{M_\ell - 1} \leq \underline{b}_{M_\ell}$ and $\underline{b}_{M^\ell} \geq \underline{b}_{M^\ell + 1}$. By symmetry we only prove the first inequality. Assume the contrary, i.e. $\underline{b}_{M_\ell - 1} > \underline{b}_{M_\ell} = \underline{a}_{M_\ell}$. Let i be the smallest index from $[m^\ell, M_\ell - 1]$ such that $\underline{b}_i = \underline{b}_{i+1} = \cdots = \underline{b}_{M_\ell - 1}$. Then $\underline{b}_i > \underline{a}_{M_\ell}$.
Case 1. $i = m^\ell$. If $\ell = 1$ we immediately get the contradiction $0 > \underline{a}_{M_\ell}$. Thus, let $\ell > 1$. Then $\underline{b}_{m_\ell} = \overline{a}_{m_\ell} > \underline{a}_{M_\ell}$, a contradiction to Lemma 9.
Case 2. $i > m^\ell$. Then $\underline{b}_{i-1} < \underline{b}_i = \underline{a}_i > \underline{a}_{M_\ell}$ in contradiction to Lemma 9.

3. Now we know that $\underline{\boldsymbol{b}}$ is feasible and conform to the min-max sequence. By construction it is obvious that if a vector $\boldsymbol{b}$ is feasible and conform to the min-max-sequence, it must satisfy $\underline{\boldsymbol{b}} \leq \boldsymbol{b} \leq \overline{\boldsymbol{b}}$. As we have shown that such a vector $\boldsymbol{b}$ exists, for instance $\boldsymbol{b} = \underline{\boldsymbol{b}}$, we can conclude $\underline{\boldsymbol{b}} \leq \overline{\boldsymbol{b}}$. This implies the feasibility of $\overline{\boldsymbol{b}}$, as Case 1 and 2 from the proof of the feasibility of $\underline{\boldsymbol{b}}$ are exactly the same, Case 3 follows from $\underline{a}_j \leq \underline{b}_j \leq \overline{b}_j \leq \overline{a}_j$ and Case 4 is analogous. The min-max-conformity of $\overline{\boldsymbol{b}}$ also follows, as $\overline{b}_{m^\ell} > \overline{b}_{m^\ell + 1}$ or $\overline{b}_{m_\ell} > \overline{b}_{m_\ell - 1}$ would again imply that there is no feasible $\boldsymbol{b}$ that is conform to the min-max sequence (as we choose $\overline{b}_{m^\ell + 1}$ and $\overline{b}_{m_\ell - 1}$ as large as possible).
Thus, $\underline{\boldsymbol{b}}$ and $\overline{\boldsymbol{b}}$ are feasible and conform to the min-max sequence. Lemma 11 yields that they (and with them all vectors $\boldsymbol{b}$ with $\underline{\boldsymbol{b}} \leq \boldsymbol{b} \leq \overline{\boldsymbol{b}}$ that are conform to the min-max sequence) are optimal solutions for the problem **Approx-MIN-DT-Row**, because they realize the lower bound for the delivery time $c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(\boldsymbol{a}) = \underline{a}_{M_1} + \sum_{\ell=2}^{t} (\underline{a}_{M_\ell} - \overline{a}_{m_\ell})$. □

We note that the case $\min\{\overline{a}_k \ : \ k \in [n]\} \geq \max\{\underline{a}_k \ : \ k \in [n]\}$ is more or less trivial, as this means that the intersection of all intervals $[\underline{a}_j, \overline{a}_j]$ is nonempty. One optimal solution is the constant vector whose value is the starting value of the above mentioned nonempty interval. Another possibility is to delete from the min-max sequence the local extrema $M^1, m_2, m^2, M_3, \ldots, m_{t-1}, m^{t-1}, M_t$ and set $t = 1$ so that we obtain

$$\boldsymbol{s} = (0 = m_1 = m^1, M_1, M^1, m_2 = m^2 = n + 1).$$

Here M_1 and M^1 are the first and last local maximum for $\boldsymbol{a}$, respectively. Let $\underline{\boldsymbol{b}}$ and $\overline{\boldsymbol{b}}$ be defined for this sequence in the same way as above. Then it is easy to check that

Theorem 9 remains true also for this case.
Since the min-max sequence as well as the vectors $\underline{\boldsymbol{b}}$ and $\overline{\boldsymbol{b}}$ can be computed in time $O(n)$, the whole problem **Approx-MIN-DT-Row** can be solved in time $O(n)$.

5.1.2 Solution of Approx-MIN-DT-TC-Row

Theorem 9 implies that a vector $\boldsymbol{b}$ is an optimal solution for the problem **Approx-MIN-DT-TC-Row** iff $\boldsymbol{b}$ is conform to the min-max sequence, $\underline{\boldsymbol{b}} \leq \boldsymbol{b} \leq \overline{\boldsymbol{b}}$ and $\|\boldsymbol{b}-\boldsymbol{a}\|_1$ is minimal. Obviously, $\boldsymbol{b}$ is conform to the min-max sequence iff

$$b_j = \underline{b}_j = \overline{b}_j \text{ for all } j \in [m_\ell, m^\ell] \text{ and } [M_\ell, M^\ell], \quad \ell = 1, \ldots, t,$$

$\boldsymbol{b}$ is increasing in $[m^\ell, M_\ell]$ and decreasing in $[M^\ell, m_{\ell+1}]$, $\ell = 1, \ldots, t$. This characterization already was the same in the special model discussed in [26] with $(a_j - \delta)_+ \leq b_j \leq a_j + \delta$. There the problem **Approx-MIN-DT-TC-Row** is reduced to the **Monotone Discrete Approximation Problem**:

MDAP: Given a vector $\boldsymbol{a} = (a_1, \ldots, a_n)$ and two increasing vectors $\underline{\boldsymbol{b}} \leq \overline{\boldsymbol{b}}$ with $\underline{b}_1 = \overline{b}_1$ and $\underline{b}_n = \overline{b}_n$, find an increasing vector $\boldsymbol{b}$ such that $\underline{\boldsymbol{b}} \leq \boldsymbol{b} \leq \overline{\boldsymbol{b}}$ and $\|\boldsymbol{b} - \boldsymbol{a}\|_1$ is minimal.

This problem **MDAP** is solved for every sequence between a local minimum and the next local maximum and for the reversed sequence between a local maximum und the next local minimum using dynamic programming. The vector size n in **MDAP** corresponds to the size of these subvectors. Thus, the solution of **Approx-MIN-DT-TC-Row** in our generalized model is exactly the same as in [26] and we omit the algorithm and the proof of its optimality here. As in [26], the problem **Approx-MIN-DT-TC-Row** can be solved in time $O(\delta n)$, where in our case $\delta = \max\{\overline{a}_j - \underline{a}_j \ : \ j \in [n]\}$.

5.1.3 Solution of Approx-MIN-DT and Approx-MIN-DT-TC

Now we want to solve the approximation problems for matrices with an arbitrary number of rows. Given a matrix A with nonnegative integer entries, let as usual $\boldsymbol{a_i}$ denote the i-th row of A for $i \in [m]$. Ignoring machine-dependent constraints, we can solve the problem **Approx-MIN-DT** as in [26] independently for each row of A, i.e. we have to solve m problems **Approx-MIN-DT-Row**. It is obvious that the optimal value of the objective function of **Approx-MIN-DT** is

$$c_{\underline{A},\overline{A}}(A) = \max\{c_{\underline{\boldsymbol{a}}_i,\overline{\boldsymbol{a}}_i}(\boldsymbol{a}_i) : i \in [m]\}.$$

Using the results from Section 5.1.1, we obtain that the problem **Approx-MIN-DT** can be solved in time $O(mn)$.
Again, as in [26], for the solution of **Approx-MIN-DT-TC** it is not necessary to realize the individual minimal DT for each row. We only have to realize the bound $c_{\underline{A},\overline{A}}(A)$ for each row. This task immediately leads us to the following **Constrained-DT and MIN-TC problem for single rows**:

CDTMTC-Row: Given the vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$, $\underline{\boldsymbol{a}} = (\underline{a}_1, \ldots, \underline{a}_n)$ and $\overline{\boldsymbol{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ with nonnegative integral entries and a bound C, find a vector $\boldsymbol{b}$ with nonnegative integral entries such that $\underline{a}_j \leq b_j \leq \overline{a}_j$ for all $j \in [n]$, $c(\boldsymbol{b}) \leq C$ and $\|\boldsymbol{a} - \boldsymbol{b}\|_1$ is minimum.

Note that if $C \leq c(\boldsymbol{a})$, we can replace $c(\boldsymbol{b}) \leq C$ by $c(\boldsymbol{b}) = C$. Now we formulate an important lemma and omit the proof as it is exactly the same as that for the corresponding lemma in [26].

Lemma 12. *Let $\boldsymbol{b}$ be an optimal solution of the problem* **CDTMTC-Row**. *We have for every $j \in [n-1]$*

$$(b_{j+1} - b_j)_+ \leq (a_{j+1} - a_j)_+.$$

As in Section 5.1.2 we solve the problem using dynamic programming. W.l.o.g. we may assume that $C < c(\boldsymbol{a})$, because otherwise $\boldsymbol{b} = \boldsymbol{a}$ is the optimal solution. Moreover, we have $c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(\boldsymbol{a}) \leq C$, because we take $C = c_{\underline{A},\overline{A}}(A)$. For our dynamic programming approach we consider the following class of problems:

CDTMTC-Row$_{i,j,k}$: Given the vectors $(a_1, \ldots, a_j)$, $(\underline{a}_1, \ldots, \underline{a}_j)$ and $(\overline{a}_1, \ldots, \overline{a}_j)$ with nonnegative integral entries, find a vector $(b_1, \ldots, b_j)$ with nonnegative integral entries such that $\underline{a}_\ell \leq b_\ell \leq \overline{a}_\ell$ for all $\ell \in [j]$, $(b_{\ell+1} - b_\ell)_+ \leq (a_{\ell+1} - a_\ell)_+$ for all $\ell \in [j-1]$, $b_j = a_j + i$, $c((a_1, \ldots, a_j)) - c((b_1, \ldots, b_j)) \geq k$ and $\sum_{\ell=1}^{j} |a_\ell - b_\ell|$ is minimum.

Let briefly

$$c(j) = c((a_1, \ldots, a_j)),$$
$$c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(j) = c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}((a_1, \ldots, a_j)).$$

Note that the values $c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(j)$ can be computed inside Algorithm 9 from the appendix using Theorem 9.
Our aim is to compute for $j \in [n]$, $i \in [-(a_j - \underline{a}_j), \overline{a}_j - a_j]$, and $k \in [0, c(j) - c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(j)]$ the minimal value $p_{i,j,k}$ of the objective function and the value $q_{i,j,k} = b_{j-1} - a_{j-1}$ for some optimal solution $(b_1, \ldots, b_j)$ of **CDTMTC-Row$_{i,j,k}$**. Here we put $p_{i,j,k} = q_{i,j,k} = \infty$ if there is no feasible solution.
Note that

$$(a + i)_+ \leq a_+ \text{ iff } i \leq (-a)_+.$$

Accordingly, for $b_j = a_j + i_1$ and $b_{j+1} = a_{j+1} + i_2$

$$(b_{j+1} - b_j)_+ \leq (a_{j+1} - a_j)_+ \text{ iff } i_2 - i_1 \leq (a_j - a_{j+1})_+. \tag{5.10}$$

Theorem 10. *Algorithm 10 from the appendix computes an optimal solution of the problem* **CDTMTC-Row** *in time $O(\delta^3 n^2)$, where $\delta = \max\{\overline{a}_j - \underline{a}_j \ : \ j \in [n]\}$.*

Again, we omit the proof as only slight changes from the proof in [26] are needed.

From Theorem 10 and the remarks at the beginning of this section it finally follows: The problem **Approx-MIN-DT-TC** can be solved in time $O(\delta^3 mn^2)$, where $\delta = \max\{\overline{a}_{ij} - \underline{a}_{ij} \ : \ (i,j) \in [m] \times [n]\}$.

5.2 Delivery time minimization with interleaf collision constraint

The unconstrained problem from the previous section is now considered in a constrained version where the interleaf collision constraint is taken into account. Recall that this constraint forbids that a left (respectively right) leaf overlaps with an adjacent right (respectively left) leaf, i.e. we require $\ell_i \leq r_{i+1} + 1$ and $r_i + 1 \geq \ell_{i+1}$ for all $i \in [m-1]$.

Example 6. For the decomposition into ICC-segments

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

we need a delivery time of 2, whereas $c(A) = 1$.

The exact decomposition problem **MIN-DT** for $\mathcal{S}' = \mathcal{S}_{ICC}$ can be solved by several efficient algorithms [6, 39, 47]. The idea underlying one of these algorithms is reviewed below, because it is the basis for our approach to the problem **Approx-MIN-DT**. Finally, we observe that the second part of each of the problems **Approx-MIN-DT** and **Approx-MIN-DT-TC**, the search for the shape matrix decomposition, can be ignored safely, because, once the matrix B is fixed, we can apply any exact decomposition algorithm to complete the task. We present a graph-theoretical characterization of the minimal DT of an approximation with a constructive proof, and show how the total change can be reduced heuristically.

5.2.1 Review of the exact decomposition

The basis of our approach is a characterization of the minimal DT of a decomposition with ICC as the maximal weight of a q-s-path in the following digraph $G = (V, E)$ [39, 41].

$$\begin{aligned} V &= \{q, s\} \cup ([m] \times [0, n+1]), \\ E &= \{(q, (i, 0)) \ : \ i \in [m]\} \cup \{((i, n+1), s) \ : \ i \in [m]\} \\ &\quad \cup \{((i, j), (i, j+1)) \ : \ i \in [m], \ j \in [0, n]\} \\ &\quad \cup \{((i, j), (i+1, j)) \ : \ i \in [m-1], \ j \in [n]\} \\ &\quad \cup \{((i, j), (i-1, j)) \ : \ i \in [2, m], \ j \in [n]\}. \end{aligned}$$

In order to avoid case distinctions, we add two columns to our matrix and put

$$a_{i0} = a_{i,n+1} = 0 \qquad \forall i \in [m].$$

Now we can define arc weights by

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

We call this graph the *DT-ICC-graph* for A. Figure 5.3 shows the DT-ICC-graph for the matrix

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

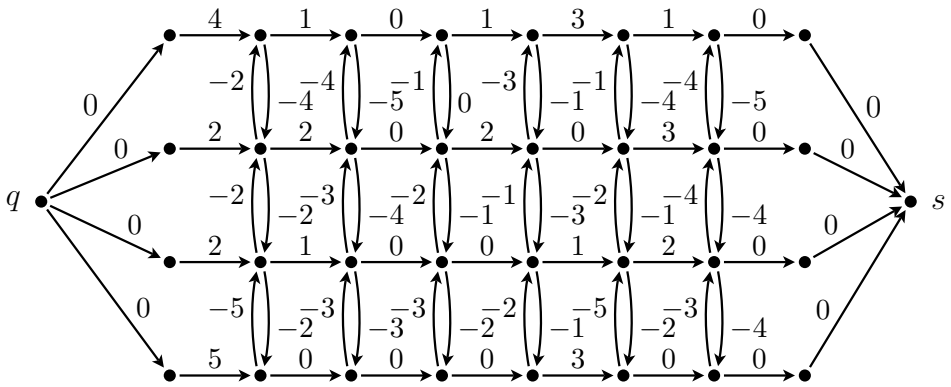

Fig. 5.3: The DT-ICC-graph for matrix A.

Definition 5 (ICC-complexity)**.** Let A be an intensity matrix, and let G be the DT-ICC-graph for A. The maximal weight of a q-s-path in G is called *ICC-complexity* of A and denoted by $c^{ICC}(A)$. More formally,

$$c^{ICC}(A) = \max\{w(P) \ : \ P \text{ is a } q\text{-}s\text{-path in } G.\}.$$

Using this definition the main result of [39] can be formulated as follows.

Theorem 11 (Kalinowski)**.** *The minimal DT of a decomposition of A with ICC equals $c^{ICC}(A)$.*

5.2.2 Approximation

To simplify our notation, for each $(i,j) \in [m] \times [n]$ we introduce the interval of acceptable fluence values

$$I_{ij} = \left[\underline{a_{ij}}, \overline{a_{ij}}\right], \qquad \underline{a_{ij}} \le a_{ij} \le \overline{a_{ij}}.$$

We want to find a matrix B such that

$$b_{ij} \in I_{ij} \ \text{ for } (i,j) \in [m] \times [n] \quad \text{and} \quad c^{ICC}(B) \to \min.$$

We follow an approach from [26] and replace every vertex $(i,j) \in [m] \times [n]$ by $|I_{ij}|$ copies, i.e. by the set

$$V_{ij} = \{(i,j)\} \times I_{ij}.$$

In order to avoid case distinctions in the discussion below we also replace the vertices in columns 0 and $n+1$ by

$$V_{i0} = \{(i,0,0)\} \quad \text{and} \quad V_{i,n+1} = \{(i,n+1,0)\}.$$

An arc $((i,j),(i,j+1))$ in the DT-ICC-graph G is replaced by the complete bipartite graph $V_{ij} \times V_{i,j+1}$, and similarly for the arcs $((i,j),(i\pm 1,j))$. The weights of the arcs $((i,j,k),(i,j+1,\ell))$ should model the approximation matrix B if we choose $b_{ij}=k$ and $b_{i,j+1}=\ell$, and similarly for the other arc types. Hence we define the arc weights by

$$\begin{aligned}
w(q,(i,0,0)) &= 0 && \forall i \in [m],\\
w((i,n+1,0),s) &= 0 && \forall i \in [m],\\
w((i,0,0),(i,1,k)) &= k && \forall i \in [m], k \in I_{i1},\\
w((i,n,k),(i,n+1,0)) &= 0 && \forall i \in [m], k \in I_{in},\\
w((i,j-1,k),(i,j,\ell)) &= (\ell-k)_+ && \forall i \in [m], j \in [n], k \in I_{i,j-1}, \ell \in I_{ij},\\
w((i,j,k),(i+1,j,\ell)) &= -k && \forall i \in [m-1], j \in [n], k \in I_{ij}, \ell \in I_{i+1,j},\\
w((i,j,k),(i-1,j,\ell)) &= -k && \forall i \in [2,m], j \in [n], k \in I_{ij}, \ell \in I_{i-1,j}.
\end{aligned}$$

In order to determine the minimal complexity of an approximation matrix we compute numbers $W(i,j,k)$ such that

$$\begin{aligned}
W(i,j,k) = \max\big\{ &\min_{\ell} W(i,j-1,\ell) + (k-\ell)_+,\\
&\min_{l} W(i-1,j,\ell) - \ell,\ \min_{l} W(i+1,j,\ell) - \ell \big\}.
\end{aligned}$$

The intuitive idea is that for every feasible approximation B with $b_{ij}=k$, the maximal weight of a q-(i,j)-path in the DT-ICC-graph for B is at least $W(i,j,k)$. The numbers $W(i,j,k)$ can be computed efficiently (complexity $O(m^2n\Delta^2)$, where Δ denotes any upper bound for $|I_{ij}|$) as described in Algorithm 3. Again, in order to avoid case distinctions at the boundaries, we add the values

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

By construction, for any feasible approximation B with $b_{in}=k$, the DT-ICC-graph for B contains a path of weight at least $W(i,n,k)$. Hence, the numbers $W(i,n,k)$ can be used to define a lower bound $c^{ICC}_{\underline{A},\overline{A}}(A)$ for the optimal value of the objective function of **Approx-MIN-DT** for $\mathcal{S}' = \mathcal{S}_{ICC}$.

Definition 6 (ICC-approximation complexity)**.** The *ICC-approximation complexity* of A (with respect to the given intervals I_{ij}) is defined by

$$c^{ICC}_{\underline{A},\overline{A}}(A) = \max_i \min_k W(i,n,k).$$

We will show that this bound is sharp by an explicit construction of an approximation matrix B with this ICC-complexity. For the last column we put

$$b_{in} = \begin{cases} a_{in} & \text{if } W(i,n,a_{in}) \le c^{ICC}_{\underline{A},\overline{A}}(A),\\ \max\{k \ :\ W(i,n,k) \le c^{ICC}_{\underline{A},\overline{A}}(A)\} & \text{otherwise.}\end{cases}$$

For $j<n$, we assume that the entries $b_{i,j+1}$ are already determined, and put

$$b_{ij} = \max\left\{k \ :\ W(i,j,k) + (b_{i,j+1}-k)_+ \le W(i,j+1,b_{i,j+1})\right\}.$$

Algorithm 3 Computation of the numbers $W(i,j,k)$

for $i \in [m]$ **do**
  $W(i,0,0) = 0$
end for
for $j = 1$ to n **do**
  for $i \in [m]$ **do**
    for all k **do**
      $W(i,j,k) = \min_\ell W(i,j-1,\ell) + (k-\ell)_+$
    end for
  end for
  for $i = 2$ to m **do**
    for all k **do**
      $W(i,j,k) = \max\left\{W(i,j,k), \min_\ell W(i-1,j,\ell) - \ell\right\}$
    end for
    for $i' = i-1$ **downto** 1 **do**
      for all k **do**
        $W(i',j,k) = \max\left\{W(i',j,k), \min_\ell W(i'+1,j,\ell) - \ell\right\}$
      end for
    end for
  end for
end for

Example 7. We consider the following fluence matrix A with $c^{ICC}(A) = 8$.

$$A = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

We choose the upper and lower bound such that $|b_{ij} - a_{ij}| \leq 1$ for every (i,j). The intervals and an optimal approximation are

$$\begin{pmatrix} [3,5] & [0,1] & [0,1] \\ [0,1] & [0,1] & [3,5] \end{pmatrix}, \qquad B = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 1 & 3 \end{pmatrix}$$

with $c^{ICC}(B) = 4$, realized by the optimal decomposition

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Our algorithm obtains matrix B as follows. First, we compute the numbers $W(i,j,k)$, and obtain, for each (i,j), a vector

$$\left(W_{i,j,\underline{a_{ij}}}, W_{i,j,\underline{a_{ij}}+1}, \ldots, W_{i,j,\overline{a_{ij}}}\right).$$

These vectors are collected in the following array:

$$\begin{matrix} (3,4,5) & (3,3) & (3,3) \\ (0,1) & (2,2) & (4,5,6) \end{matrix}.$$

Thus, the optimal DT is

$$\max\{\min\{3,3\},\min\{4,5,6\}\} = 4.$$

For the third column, we choose $b_{13} = 0$ and $b_{23} = 3$. For the entry $(1,2)$ we have

$$W(1,2,0) + w((1,2,0),(1,3,0)) = W(1,2,1) + w((1,2,1),(1,3,0)) = W(1,3,0).$$

We choose the maximal possible value $b_{12} = 1$. Observe that $b_{12} = 0$ is indeed not possible, since it leads to an increased DT. For entry $(2,2)$, we have

$$W(2,2,0) + w((2,2,0),(2,3,3)) = 2 + 3 > W(2,3,3),$$

so here $b_{22} = 1$ is the only possible choice. Similarly, we get $b_{11} = 3$ and $b_{21} = 1$. Clearly, the latter one can be replaced by 0.

In order to prove that our method is correct, we need some simple properties of the numbers $W(i,j,k)$.

Lemma 13. *For every $(i,j) \in [m] \times [n]$ and every k such that $(i,j,k),(i,j,k+1) \in V_{ij}$ we have*

$$W(i,j,k) \le W(i,j,k+1) \le W(i,j,k) + 1. \tag{5.11}$$

Furthermore, $W(i,j,k+1) = W(i,j,k) + 1$ iff

$$W(i,j,k) = W(i,j-1,\ell) + (k-\ell)_+$$

for some $\ell \in I_{i,j-1}$ with $\ell \le k$.

Proof. Since

$$W(i,j-1,\ell) + (k-\ell)_+ \le W(i,j-1,\ell) + (k+1-\ell)_+$$

and using the definition of the $W(i,j,k)$, we conclude $W(i,j,k) \le W(i,j,k+1)$. On the other hand, we have

$$\begin{aligned} W(i,j,k) &= \max\big\{ \min_\ell W(i,j-1,\ell) + (k-\ell)_+, \\ &\qquad \min_\ell W(i-1,j,\ell) - \ell,\ \min_\ell W(i+1,j,\ell) - \ell \big\} \\ &\ge \max\big\{ \min_\ell W(i,j-1,\ell) + (k+1-\ell)_+, \\ &\qquad \min_\ell W(i-1,j,\ell) - \ell,\ \min_\ell W(i+1,j,\ell) - \ell \big\} - 1 \\ &= W(i,j,k+1) - 1, \end{aligned}$$

where equality occurs iff $W(i,j,k) = W(i,j-1,l) + (k-\ell)_+$ and $k \ge \ell$. □

The next lemma is the key step of our argumentation. It asserts that the chosen b_{ij} do not lead to conflicts inside the columns.

Lemma 14. *For all j and all $i \in [m-1]$, we have*

$$W(i,j,b_{ij}) - b_{ij} \leq W(i+1,j,b_{i+1,j}),$$

and for all j and all $i \in [2,m]$, we have

$$W(i,j,b_{ij}) - b_{ij} \leq W(i-1,j,b_{i-1,j}).$$

Proof. We only show the first statement, since the second one can be proved similarly. Suppose the statement is wrong, i.e.

$$W(i,j,b_{ij}) - b_{ij} > W(i+1,j,b_{i+1,j}).$$

By construction, there is some $k \in I_{ij}$ such that $W(i,j,k) - k \leq W(i+1,j,b_{i+1,j})$.
Case 1. $k < b_{ij}$. Let $\delta = b_{ij} - k > 0$. By Lemma 13, we have

$$W(i,j,k) \geq W(i,j,b_{ij}) - \delta.$$

But now we obtain

$$W(i,j,k) - k \geq (W(i,j,b_{ij}) - \delta) - (b_{ij} - \delta) > W(i+1,j,b_{i+1,j}),$$

and this is the required contradiction.
Case 2. $k > b_{ij}$. Let $\delta = k - b_{ij} > 0$. By construction of the numbers b_{ij},

$$\begin{aligned} W(i,j,b_{ij}) + (b_{i,j+1} - b_{ij})_+ &\leq W(i,j+1,b_{i,j+1}), \\ W(i,j,b_{ij}+1) + (b_{i,j+1} - (b_{ij}+1))_+ &> W(i,j+1,b_{i,j+1}). \end{aligned}$$

Using Lemma 13, this is possible only if

$$W(i,j,b_{ij}+1) = W(i,j,b_{ij}) + 1.$$

Using Lemma 13 repeatedly, we obtain

$$W(i,j,k) = W(i,j,b_{ij}) + \delta.$$

But together this implies $W(i,j,k) - k = W(i,j,b_{ij}) - b_{ij}$, which is a contradiction.□

Now let G be the DT-ICC-graph for B. Denote by $\alpha_1(i,j)$ the maximal weight of a q-(i,j)-path in G. Note that the numbers $\alpha_1(i,j)$ can be computed similarly to the numbers $W(i,j,k)$. Clearly, $\alpha_1(i,1) = b_{i1}$, and the procedure for column $j \geq 2$ is described in Algorithm 4.

Lemma 15. *For all (i,j) we have $\alpha_1(i,j) \leq W(i,j,b_{ij})$.*

Proof. We use induction on j. For $j = 1$ the claim is obvious:

$$\alpha_1(i,1) = W(i,1,b_{i1}) = b_{i1}.$$

Algorithm 4 Computation of the numbers $\alpha_1(i,j)$ for fixed $j \geq 2$

for $i \in [m]$ **do**
  $\alpha_1(i,j) = \alpha_1(i,j-1) + (b_{ij} - b_{i,j-1})_+$
end for
for $i = 2$ **to** m **do**
  $\alpha_1(i,j) = \max\{\alpha_1(i,j), \alpha_1(i-1,j) - b_{i-1,j}\}$
end for
for $i' = i-1$ **downto** 1 **do**
  $\alpha_1(i',j) = \max\{\alpha_1(i',j), \alpha_1(i'+1,j) - b_{i'+1,j}\}$
end for

Now let $j > 1$. After the initialization of the numbers $\alpha_1(i,j)$ in the first loop of Algorithm 4 above, we obtain for every i

$$\alpha_1(i,j) = \alpha_1(i,j-1) + (b_{ij} - b_{i,j-1})_+ \\ \leq W(i,j-1,b_{i,j-1}) + (b_{ij} - b_{i,j-1})_+ \leq W(i,j,b_{ij}).$$

We just have to check that this inequalities remain valid in every updating step. Suppose the first violation occurs when we replace $\alpha_1(i,j)$ by $\alpha_1(i \pm 1, j) - b_{i\pm1,j}$. In this case,

$$\alpha_1(i,j) = \alpha_1(i \pm 1, j) - b_{i\pm1,j} \leq W(i \pm 1, j, b_{i\pm1,j}) - b_{i\pm1,j} \leq W(i,j,b_{ij}),$$

where the last inequality is Lemma 14. So the statement of the lemma remains valid after the updating step. □

By Lemma 15 (and Theorem 11), matrix B allows a decomposition with $DT \leq c^{ICC}_{\underline{A},\overline{A}}(A)$ and this implies the following theorem.

Theorem 12. *The optimal value of the objective function of* ***Approx-MIN-DT*** *for* $\mathcal{S}' = \mathcal{S}_{ICC}$ *is* $c^{ICC}_{\underline{A},\overline{A}}(A)$ *and an approximation matrix* B *realizing this delivery time can be constructed as described above in time* $O(m^2 n \Delta^2)$.

Proof. The only thing that is left to prove is the complexity statement. For this it is sufficient to note that the computation of the numbers $W(i,j,k)$ dominates the computation time, since this has complexity $O(m^2 n \Delta^2)$ as can be seen immediately from Algorithm 3. But after the numbers $W(i,j,k)$ have been computed, we look at every entry (i,j) only once and in order to fix b_{ij} we have to do at most $|I_{ij}|$ comparisons. So the matrix B is determined in time $O(mn\Delta)$ and this concludes the proof. □

5.2.3 Reducing the total change

Now we present a heuristic approach for **Approx-MIN-DT-TC**. The construction described in Section 5.2.2 leads to an approximation B with minimal delivery time, but a large total change $\|A - B\|_1$. The reason is, that we put

$$b_{ij} = \max\{k \ : \ W(i,j,k) + (b_{i,j+1} - k) \leq W(i,j+1,b_{i,j+1})\},$$

even if none of the vertices (i, j, k) is critical, i.e. part of a q-s-path of maximal weight in the DT-ICC-graph of a feasible approximation of A. Thus, the aim is to find an approximation with the same delivery time, but smaller total change. Clearly, we can replace b_{ij} by a value b'_{ij} with $b_{ij} < b'_{ij} \leq a_{ij}$ in the case $b_{ij} < a_{ij}$, respectively with $a_{ij} \leq b'_{ij} < b_{ij}$ in the case $b_{ij} > a_{ij}$, if this decision does not increase the maximal weight of a q-s-path in the DT-ICC-graph of B.
Let therefore G be the DT-ICC-graph of B and let $\alpha_1(i, j)$ denote the maximal weight of a q-(i, j)-path in G. Similarly, let $\alpha_2(i, j)$ denote the maximal weight of an (i, j)-s-path in G. The values $\alpha_2(i, j)$ can be computed similarly to the numbers $\alpha_1(i, j)$.

Definition 7 ((i, j)-feasible)**.** Let B with $b_{ij} \in I_{ij}$ for all $(i, j) \in [m] \times [n]$ be given. For $(i, j) \in [m] \times [n]$, an integer b is called $(i, j)-$*feasible* (with respect to B) if the following conditions are satisfied:

1. $b \in I_{ij}$
2. $\alpha_1(i, j-1) + (b - b_{i,j-1})_+ + (b_{i,j+1} - b)_+ + \alpha_2(i, j+1) \leq c^{ICC}_{\underline{A},\overline{A}}(A)$
3. $i = 1$ or $\alpha_1(i, j-1) + (b - b_{i,j-1})_+ - b + \alpha_2(i-1, j) \leq c^{ICC}_{\underline{A},\overline{A}}(A)$
4. $i = m$ or $\alpha_1(i, j-1) + (b - b_{i,j-1})_+ - b + \alpha_2(i+1, j) \leq c^{ICC}_{\underline{A},\overline{A}}(A)$
5. $i = 1$ or $\alpha_1(i-1, j) - b_{i-1,j} + (b_{i,j+1} - b)_+ + \alpha_2(i, j+1) \leq c^{ICC}_{\underline{A},\overline{A}}(A)$
6. $i = m$ or $\alpha_1(i+1, j) - b_{i+1,j} + (b_{i,j+1} - b)_+ + \alpha_2(i, j+1) \leq c^{ICC}_{\underline{A},\overline{A}}(A)$
7. $i \in \{1, m\}$ or $\alpha_1(i-1, j) - b_{i-1,j} - b + \alpha_2(i+1, j) \leq c^{ICC}_{\underline{A},\overline{A}}(A)$
8. $i \in \{1, m\}$ or $\alpha_1(i+1, j) - b_{i+1,j} - b + \alpha_2(i-1, j) \leq c^{ICC}_{\underline{A},\overline{A}}(A)$

In other words, b is (i, j)-feasible iff we can replace b_{ij} by b without destroying the DT-optimality of B. Figure 5.4 illustrates the different possibilities for a path to pass through vertex (i, j). Each of these possibilities corresponds to one of the conditions 2 through 8 in Definition 7.
We propose a heuristics, formally described in Algorithm 5, to reduce the total change. Clearly, the application of this algorithm can be iterated until no more changes occur. This heuristics for minimizing the total change runs efficiently and computations show, that it finds near-optimal solutions. An exact solution of the problem **Approx-MIN-DT-TC** for ICC-segments can be found in [44].
Finally, we want to provide some numerical results for the problem **Approx-MIN-DT** both for the unconstrained case (using the algorithm from Section 5.1) and for the case $\mathcal{S}' = \mathcal{S}_{ICC}$. We again use the 475 clinical intensity matrices provided by the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf. In our tests, we choose the upper and lower bounds for the entries such that each entry is changed by at most 2, i.e. we put

$$\underline{a_{ij}} = (a_{ij} - 2)_+, \qquad \overline{a_{ij}} = a_{ij} + 2.$$

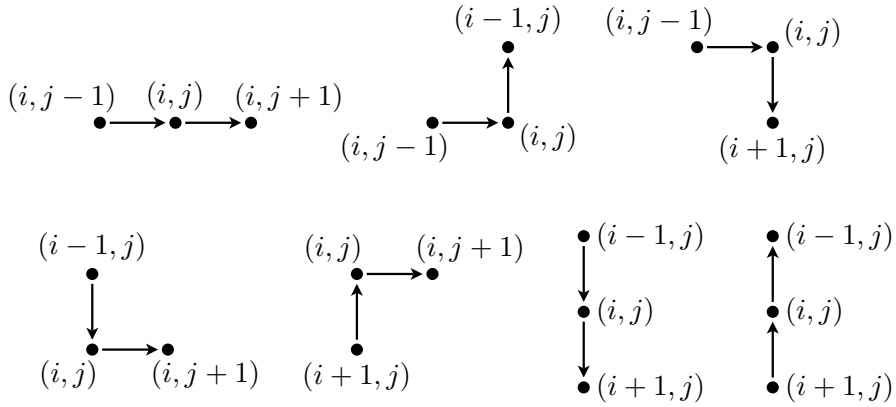

Fig. 5.4: The seven different path types that are affected by the choice of b_{ij}.

Algorithm 5 Heuristics for total change minimization

```
for j = 1 to n do
  for i = 1 to m do
    if b_ij < a_ij and b_ij + 1 is (i, j)−feasible then
      b_ij = b_ij + 1
    end if
    if b_ij > a_ij and b_ij − 1 is (i, j)−feasible then
      b_ij = b_ij − 1
    end if
    Update the numbers α_1(k, l) and α_2(k, l)
  end for
end for
```

We provide the following averaged quantities: the number of columns m of the matrices, the number of rows n of the matrices, the delivery time of the exact decomposition $c(A)$, the delivery time of the exact ICC-decomposition $c^{ICC}(A)$, the optimal delivery time of unconstrained approximate decomposition $c_{\underline{A},\overline{A}}(A)$, the optimal total change for unconstrained approximate decomposition TC_1, the optimal delivery time of approximate ICC-decompositions $c^{ICC}_{\underline{A},\overline{A}}(A)$, the total change of approximate ICC-decompositions TC_2 according to our algorithm from Section 5.2.2, the improved value of the total change TC_3 using the heuristics from Algorithm 5 and finally the optimal value of the total change TC_4 that was computed using the ideas from [44] (there, the author reduces the problem to a minimum cost flow problem). The results are shown in Table 5.1.

m	n	$c(A)$	$c^{ICC}(A)$	$c_{\underline{A},\overline{A}}(A)$	TC_1	$c^{ICC}_{\underline{A},\overline{A}}(A)$	TC_2	TC_3	TC_4
19.47	20.76	39.41	43.92	26.07	65.08	26.77	626.75	103.48	81.04

Tab. 5.1: Average test results for Approx-MIN-DT for the unconstrained case and for ICC-decompositions.

Our algorithms are completely practicable, e.g. the results for ICC-decompositions using the approaches presented in this section could be produced within a minute on a 2.5GHz workstation. Basically, we can draw four conclusions from our results.

1. The delivery time for exact decompositions into ICC-segments is slightly larger than in the unconstrained case.

2. The approximation approach leads to a significant DT-reduction: Allowing a change of at most 2 for each entry reduces the DT by more than 30% both in the unconstrained and in the constrained case.

3. For the approximation problems, the delivery time is again only a little larger in the constrained case.

4. Our heuristics from Algorithm 5 leads to a large total change reduction. The total change realized by the simple heuristics is close to the optimal total change and it is not necessary to solve minimum cost flow problems.

6. APPROXIMATE DISCRETE SEGMENTATION FOR TOTAL CHANGE MINIMIZATION

This section deals with the problem **Approx-MIN-TC**: Given the target matrix A and a set of feasible segments $\mathcal{S}' \subseteq \mathcal{S}$, find a nonnegative integer linear combination $B = \sum_{S \in \mathcal{S}'} u_S S$ such that $\|B - A\|_1$ is minimum. We start with a thorough discussion of an even more general vector approximation problem and provide its relations to and applications in IMRT. Afterwards, we present a column-generation approach for the problem **Approx-MIN-TC** that will be of major interest also for Section 6.4, where we introduce a clinically applicable segmentation algorithm.

6.1 The problem Approx-MIN-TC in general

We now discuss the problem **Approx-MIN-TC** for an arbitrary, not specified subset of segments $\mathcal{S}'$. And actually, we will even discuss a more general problem and look for a decomposable approximation B satisfying

$$\|B - A\|_\infty := \max_{(i,j) \in [m] \times [n]} |a_{ij} - b_{ij}| \leq C \tag{6.1}$$

for some given constant $C \in \mathbb{Z}_+ \cup \{\infty\}$ (possibly, such a matrix B does not exist) that minimizes

$$\|B - A\|_1 = \sum_{(i,j) \in [m] \times [n]} |a_{ij} - b_{ij}|. \tag{6.2}$$

The constraint (6.1) aims at avoiding large bixel-wise differences between target fluence A and realized fluence B (that might lead to undesirable hot spots in the treatment), and the objective (6.2) measures the total change in fluence with respect to the intensity matrix. The results of this section are based on the publication of Engelbeen, Fiorini and Kiesel [29].

The approximation problem described above motivates the definition of the following **Closest Vector Problem (CVP)**:

Input: A collection $\mathcal{G} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k\}$ of binary vectors in $\{0,1\}^d$ (the *generators*), a vector $\boldsymbol{a}$ in $\mathbb{Z}_+^d$ (the *target* vector) and an upper bound C in $\mathbb{Z}_+ \cup \{\infty\}$.

Goal: Among all vectors $\boldsymbol{b} := \sum_{j=1}^k u_j \boldsymbol{g}_j$ with $u_j \in \mathbb{Z}_+$ for $j \in [k]$, find one satisfying $\|\boldsymbol{a} - \boldsymbol{b}\|_\infty \leqslant C$ and furthermore minimizing $\|\boldsymbol{a} - \boldsymbol{b}\|_1$. If all such vectors $\boldsymbol{b}$ satisfy $\|\boldsymbol{a} - \boldsymbol{b}\|_\infty > C$, report that the instance is infeasible.

Measure: The *total change* $TC := ||\boldsymbol{a} - \boldsymbol{b}||_1$.

Considering the matrices of **Approx-MIN-TC** as vectors of size mn yields a special instance of the **CVP**. The **CVP** for $C = \infty$ exactly corresponds to the previous problem **Approx-MIN-TC** then. We remark that this **CVP** differs significantly from the intensively studied **CVP** on a lattice that is used in cryptography (see, for instance, the recent survey by Micciancio and Regev [57]).
In order to cope with the NP-hardness of the **CVP**, we design (polynomial-time, bi-criteria) approximation algorithms. For the version of the **CVP** studied here it is natural to consider approximation algorithms with *additive* approximation guarantees. We say that a polynomial-time algorithm is a $(\Delta_\infty, \Delta_1)$-*approximation algorithm* for the **CVP** if it either proves that the given instance has no feasible solution, or returns a vector $\boldsymbol{b} = \sum_{j=1}^{k} u_j \boldsymbol{g}_j$ with $u_j \in \mathbb{Z}_+$ for $j \in [k]$ such that $\|\boldsymbol{a} - \boldsymbol{b}\|_\infty \leqslant C + \Delta_\infty$ and $\|\boldsymbol{a} - \boldsymbol{b}\|_1 \leqslant OPT + \Delta_1$, where OPT is the cost of an optimal solution[1]. Notice that we cannot expect such an approximation algorithm to always either prove that the given instance is infeasible or return a feasible solution, because deciding whether an instance is feasible or not is NP-complete (this claim holds even when C is a small constant).
This section is organized as follows: We start by observing in Section 6.1.1 that the particular case where the generators form a totally unimodular matrix is solvable in polynomial time. We also provide a direct reduction to minimum cost flow when the generators have the consecutive ones property. This problem corresponds to the single row case of the problem **Approx-MIN-TC**. We also solve the problem **Approx-MIN-TC** for $\mathcal{S}' = \mathcal{S}_{MSC}$ (recall the definition of this set from Section 3) and show that this is just a special case of the one row problem. We afterwards show in Section 6.1.2 that, when $\mathcal{G}$ is a general set of generators, for all $\varepsilon > 0$, the **CVP** admits no polynomial-time $(\Delta_\infty, \Delta_1)$-approximation algorithm with $\Delta_1 \leqslant (\ln 2 - \varepsilon)\, d$, unless P = NP. (This in particular implies that the **CVP** is NP-hard.) We provide a further hardness result and prove that **Approx-MIN-TC** is already NP-hard if A has two rows.
In order to cope with this NP-hardness, we go on with the analysis of a natural $(\Delta_\infty, \Delta_1)$-approximation algorithm for the problem based on randomized rounding [58], with $\Delta_\infty = O(\sqrt{d \ln d}\,)$ and $\Delta_1 = O(d\sqrt{d \ln d}\,)$. Finally, in Section 6.1.3, we discuss the incorporation of the delivery time into the objective function and of position dependant dose constraints as in Section 5.1.

6.1.1 One row case and minimum separation constraint

In this subsection we study the **CVP** under the assumption that the binary matrix formed by the generators is totally unimodular and prove that the **CVP** is polynomial in this case. Afterwards, we give a direct reduction to a minimum cost flow problem in the case that all generators have the consecutive ones property.

[1] If the instance is infeasible, then we let $OPT = \infty$.

Consider the following natural LP relaxation of the **CVP**:

$$\text{(LP)} \quad \min \sum_{i=1}^{d} (\alpha_i + \beta_i)$$

$$\text{subject to} \quad \sum_{j=1}^{k} u_j g_{ij} - \alpha_i + \beta_i = a_i \qquad \forall i \in [d] \qquad (6.3)$$

$$\alpha_i \geqslant 0 \qquad \forall i \in [d] \qquad (6.4)$$

$$\beta_i \geqslant 0 \qquad \forall i \in [d] \qquad (6.5)$$

$$\alpha_i \leqslant C \qquad \forall i \in [d] \qquad (6.6)$$

$$\beta_i \leqslant C \qquad \forall i \in [d] \qquad (6.7)$$

$$u_j \geqslant 0 \qquad \forall j \in [k] \qquad (6.8)$$

In this relaxation, the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ model the deviation between the vector $\boldsymbol{b} := \sum_{j=1}^{k} u_j \boldsymbol{g}_j$ and the target vector $\boldsymbol{a}$. In the IMRT context, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ model the positive and negative differences between realized fluence and target fluence. Clearly, an ILP formulation of the **CVP** can be obtained from (LP) by adding the integrality constraints $u_j \in \mathbb{Z}_+$ for $j \in [k]$.

Let G denote the $d \times k$ binary matrix whose columns are $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k$. If G is totally unimodular, then the same holds for the constraint matrix of (LP). Because $\boldsymbol{a}$ and C are integer, any basic feasible solution of (LP) is integer. Thus, solving the **CVP** amounts to solving (LP) when G is totally unimodular. Hence, we obtain the following easy result.

Theorem 13. *The* **CVP** *restricted to instances such that the generators form a totally unimodular matrix can be solved in polynomial time.*

For the rest of this section, assume that the generators satisfy the consecutive ones property. In particular, G is totally unimodular. This case is of special interest, because it corresponds to the one row case of the segmentation problem in the IMRT context. We show that it is not necessary to solve an LP and provide a direct reduction to a minimum cost flow problem.

We begin by appending a row of zeros to the matrix G and vector $\boldsymbol{a}$. Similarly, we add an extra row to the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Thus, the matrix and the vectors now have $d + 1$ rows. Next, we replace (6.3) by an equivalent set of equations: We keep the first equation, and replace each other equation by the difference between this equation and the previous one. Because the resulting constraint matrix is the incidence matrix of a network, we conclude that (LP) actually models a minimum cost network flow problem. We give more details below.

We denote the generators by $\boldsymbol{g}_{\ell,r}$ where $[\ell, r]$ is the interval of ones of this generator. That is, $\boldsymbol{g} = \boldsymbol{g}_{\ell,r}$ if and only if $g_i = 1$ for $i \in [\ell, r]$ and $g_i = 0$ otherwise. Let $\mathcal{I}$ be the set of intervals such that $\mathcal{G} = \{\boldsymbol{g}_{\ell,r} \mid [\ell, r] \in \mathcal{I}\}$. We assume that there is no generator with an empty interval of ones (that is, $\ell \leqslant r$ always holds). Now, let D be the network

whose set of nodes and (possibly multi-)set of arcs are respectively defined as:

$$\begin{aligned} V(D) &:= [d+1] = \{1, 2, \ldots, d+1\}, \quad \text{and} \\ A(D) &:= \big\{(i, i+1) \mid i \in [d]\big\} \cup \big\{(i+1, i) \mid i \in [d]\big\} \cup \big\{(\ell, r+1) \mid [\ell, r] \in \mathcal{I}\big\}. \end{aligned}$$

Let us notice that parallel arcs can appear when the interval of a generator only contains one element. In such a case, we keep both arcs: the one representing the generator and the other one.

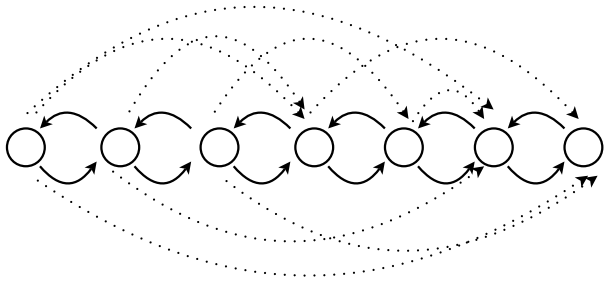

Fig. 6.1: The network for an instance with $d = 6$ and $k = 9$.

Letting $a_0 := 0$, we define the demand of each node $j \in V(D)$ as $a_{j-1} - a_j$. The arcs of type $(j, j+1)$ and $(j+1, j)$ have capacity C and cost 1. The other arcs, that is, the arcs corresponding to the generators, have infinite capacity and cost 0. An example of the network is shown in Figure 6.1. If we consider a flow ϕ in the network, we have the following correspondence between the flow values and the variables of the LP:

$$\begin{aligned} \phi(\ell, r+1) &= u_{\ell,r} \text{ for all } [\ell, r] \in \mathcal{I}, \\ \phi(i, i+1) &= \beta_i \text{ for all } i \in [d], \\ \phi(i+1, i) &= \alpha_i \text{ for all } i \in [d]. \end{aligned}$$

From the discussion above, we obtain the following result.

Theorem 14. *Let $\mathcal{G}$ and D be as above, let $\boldsymbol{a} \in \mathbb{Z}_+^d$ and $C \in \mathbb{Z}_+ \cup \{\infty\}$, and let OPT denote the optimal value of the corresponding* **CVP** *instance. Then, OPT equals the minimum cost of a flow in D.*

There are various polynomial time algorithms for minimum cost flow problems, e.g. the cost scaling algorithm has a complexity of $O(n^3 \log(nM))$ where M is the maximum cost of an arc in the network (cf. [2] for this and more algorithms). This provides a time complexity of $O(n^3 \log(n))$ for solving our problem.
Our network D is similar to the network used in [1] for finding exact unconstrained decompositions. There, the arcs of type $(j, j+1)$ and $(j+1, j)$ modeling the total change are missing and the arcs of type $(\ell, r+1)$ are available for all nonempty intervals $[\ell, r]$.

At the end of this section, we consider the problem **Approx-MIN-TC** under the constraint that the set $\mathcal{S}'$ of generators is formed by all segments that satisfy the minimum

separation constraint, i.e. $\mathcal{S}' = \mathcal{S}_{MSC}$. Given $\lambda \in [n]$, this constraint requires that the rows which are not totally closed have a leaf opening of at least λ. Mathematically, the leaf positions of open rows $i \in [m]$ have to satisfy $r_i - \ell_i \geqslant \lambda - 1$. We cannot decompose any matrix A under this constraint. Indeed, the following single row matrix cannot be decomposed for $\lambda = 3$:

$$A = \begin{pmatrix} 1 & 1 & 4 & 1 & 1 \end{pmatrix}.$$

The problem of determining if it is possible to decompose a matrix A under this constraint was proved to be polynomial by Kamath et al. [47].
Obviously, the minimum separation constraint is a restriction on the leaf openings in each single row, but does not affect the combination of leaf openings in different rows. Again, more formally, the set of allowed leaf openings in one row i is

$$\mathcal{S}_i = \{[\ell_i, r_i] \mid r_i - \ell_i \geqslant \lambda - 1 \text{ or } r_i = \ell_i - 1\},$$

and does not depend on i. If we denote a segment by the sequence of its leaf positions $([\ell_1, r_1], \ldots, [\ell_m, r_m])$, then the set of feasible segments $\mathcal{S}_{MSC}$ for the minimum separation constraint is simply $\mathcal{S}_{MSC} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$. Thus, in order to solve **Approx-MIN-TC** under the minimum separation constraint, it is sufficient to focus on single rows. Indeed, whenever the set of feasible segments has a structure of the form $\mathcal{S}' = \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$, which means that the single row solutions can be combined arbitrarily and we always get a feasible segment, solving the single row problem is sufficient. From Theorem 13, we infer our next result.

Corollary 15. *The problem* **Approx-MIN-TC** *for $\mathcal{S}' = \mathcal{S}_{MSC}$ can be solved in polynomial time.*

6.1.2 Hardness of the CVP and approximation algorithm

In this subsection we prove that the **CVP** is NP-hard to approximate within an additive error of at most $(\ln 2 - \varepsilon)d$, for all $\varepsilon > 0$. To prove this, we consider the particular case where $\boldsymbol{a}$ is the all-one vector. The given set $\mathcal{G}$ is formed of k binary vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k$. Because $\boldsymbol{a}$ is binary, the associated coefficients u_j for $j \in [k]$ can be assumed to be binary as well.
For our hardness results, we need a special type of satisfiability problem. A *3SAT-6 formula* is a conjunctive normal form (CNF formula) in which every clause contains exactly three literals, every literal appears in exactly three clauses and a variable appears at most once in each clause. This means that each variable appears three times negated and three times unnegated. Such a formula is said to be *δ-satisfiable* if at most a δ-fraction of its clauses is satisfiable.
As noted by Feige, Lovász and Tetali [33], the following result is a consequence of the PCP theorem (see Arora, Lund, Motwani, Sudan and Szegedy [3]).

Theorem 16 ([33])**.** *There is some $0 < \delta < 1$, such that it is NP-hard to distinguish between a satisfiable 3SAT-6 formula and one which is δ-satisfiable.*

By combining the above theorem and a reduction due to Feige [32] one gets the following result (see Feige, Lovász and Tetali [33] and also Cardinal, Fiorini and Joret [13]):

Lemma 16 ([13, 33])**.** *For any given constants $c > 0$ and $\xi > 0$, there is a polynomial time reduction associating to any 3SAT-6 formula Φ a corresponding set system $\mathcal{S}(\Phi) = (V, \mathscr{S})$ with the following properties:*

- *The sets of $\mathscr{S}$ all have the size d/t, where $d = |V|$ and t can be assumed to be arbitrarily large.*
- *If Φ is satisfiable, then V can be covered by t disjoint sets of $\mathscr{S}$.*
- *If Φ is δ-satisfiable, then every x sets chosen from $\mathscr{S}$ cover at most a $1 - \left(1 - \frac{1}{t}\right)^x + \xi$ fraction of the points, for $1 \leqslant x \leqslant ct$.*

Theorem 17. *For all $\varepsilon > 0$, there exists no polynomial-time $(\Delta_\infty, \Delta_1)$-approximation algorithm for the* **CVP** *with $\Delta_1 \leqslant (\ln 2 - \varepsilon)\, d \approx (0.693 - \varepsilon)d$, unless $P = NP$.*

Proof. We use Lemma 16 to obtain a reduction from 3SAT-6 to the **CVP** (by identifying subsets with their characteristic binary vectors) with the following properties: For any given constants $c > 0$ and $\xi > 0$, it is possible to set the values of the parameters of the reduction in such a way that:

- The generators from $\mathcal{G}$ all have the same number $\frac{d}{t}$ of ones, where t can be assumed to be larger than any given constant.
- If the 3SAT-6 formula Φ is satisfiable, then $\boldsymbol{a}$ can be exactly decomposed as a sum of t generators of $\mathcal{G}$.
- If the 3SAT-6 formula Φ is δ-satisfiable, then the support of any linear combination of x generators chosen from $\mathcal{G}$ is of size at most $d\left(1 - \left(1 - \frac{1}{t}\right)^x + \xi\right)$, for $1 \leqslant x \leqslant ct$.

From what precedes, if Φ is satisfiable then the **CVP** instance is feasible and $OPT = 0$. We claim that if Φ is δ-satisfiable, then *any* approximation $\boldsymbol{b} := \sum_{j=1}^{k} u_j \boldsymbol{g}_j$ with $u_j \in \mathbf{Z}_+$ for $j \in [k]$ has total change $TC := ||\boldsymbol{a} - \boldsymbol{b}||_1 > d\,(\ln 2 - \varepsilon)$, provided t is large enough and ξ is small enough (this is proved below).
The claim implies the theorem for the following reason: Let us assume there exists a polynomial-time $(\Delta_\infty, \Delta_1)$-approximation algorithm with $\Delta_1 \leqslant (\ln 2 - \varepsilon)\, d$ for the **CVP** with some nonnegative integer bound C. Moreover, assume that we are given a 3SAT-6 formula that is either satisfiable or δ-satisfiable.
The approximation algorithm either declares the instance given by the reduction to be infeasible or provides an approximation $\boldsymbol{b}$. In the first case, we can conclude that Φ is not satisfiable, hence δ-satisfiable. In the latter case, we compare the total change TC of the solution returned by the algorithm to $(\ln 2 - \varepsilon)\, d$. If $TC \leqslant (\ln 2 - \varepsilon)\, d$ then the claim implies that Φ is satisfiable. If $TC > (\ln 2 - \varepsilon)\, d$ then we can conclude that Φ is not satisfiable, hence δ-satisfiable, because otherwise the **CVP** instance would be feasible with $OPT = 0$ and the approximation returned by the algorithm should satisfy $TC \leqslant 0 + \Delta_1 \leqslant (\ln 2 - \varepsilon)\, d$. In conclusion, we could use the algorithm to decide if Φ is satisfiable or δ-satisfiable in polynomial time. By Theorem 16, this would imply P = NP, a contradiction.

Now, we prove the claim. Notice that we may assume that $u_j \in \{0,1\}$ for all $j \in [k]$. Let x be denote the number of coordinates u_j that are nonzero. We distinguish three cases.

- **Case 1:** $x = 0$**.**

  In this case $TC = d > (\ln 2 - \varepsilon)d$.

- **Case 2 :** $1 \leqslant x \leqslant ct$**.**

  Let ρ denote the number of components b_i of $\boldsymbol{b}$ that are nonzero. Thus $d - \rho$ is the number of b_i equal to 0. The total change of $\boldsymbol{b}$ includes one unit for each component of $\boldsymbol{b}$ that is zero and a certain number of units caused by components of $\boldsymbol{b}$ larger than one. More precisely, we have:

$$\begin{aligned} TC &= d - \rho + x\frac{d}{t} - \rho \\ &\geqslant d\left(\frac{x}{t} + 1\right) - 2d\left(1 - \left(1 - \frac{1}{t}\right)^x + \xi\right) \\ &= d\left(\frac{x}{t} + 2\left(1 - \frac{1}{t}\right)^x - 1 - 2\xi\right) \\ &= d\left((1-\beta)\,x + 2\beta^x - 1 - 2\xi\right), \end{aligned}$$

  where $\beta := 1 - \frac{1}{t}$. Note that $\beta < 1$ and taking t large corresponds to taking β close to 1. In order to derive the desired lower bound on the total change of $\boldsymbol{b}$ we now study the function $f(x) := (1-\beta)\,x + 2\beta^x$. The first derivative of f is

$$f'(x) = (1-\beta) + 2\ln\beta \cdot \beta^x.$$

  It is easy to verify (since the second derivative of f is always positive) that f is convex and attains its minimum at

$$x_{\min} = \frac{1}{\ln\beta} \cdot \ln\left(\frac{\beta - 1}{2\ln\beta}\right).$$

  Hence, we have for all $x > 0$,

$$\begin{aligned} f(x) &\geqslant f(x_{\min}) \\ &= (1-\beta)\,x_{\min} + 2\beta^{x_{\min}} \\ &= (1-\beta) \cdot \frac{1}{\ln\beta} \cdot \ln\left(\frac{\beta-1}{2\ln\beta}\right) + \frac{\beta-1}{\ln\beta} \\ &= \frac{\beta-1}{\ln\beta}\left(\ln\left(\frac{2\ln\beta}{\beta-1}\right) + 1\right). \end{aligned}$$

  By l'Hospital's rule,

$$\lim_{\beta\to 1} \frac{(\beta-1)}{\ln\beta} = 1,$$

hence we have

$$f(x) \geqslant \ln 2 + 1 + 2\xi - \varepsilon$$

for t sufficiently large and ξ sufficiently small, which implies

$$TC \geqslant d\,(\ln 2 - \varepsilon)\,.$$

- **Case 3:** $x > ct$**.**

  Let again ρ be the number of components b_i of $\boldsymbol{b}$ that are nonzero. The first ct generators used by the solution have some common nonzero entries. By taking into account the penalties caused by components of $\boldsymbol{b}$ larger than one, we have:

$$\begin{aligned} TC &\geqslant ct \cdot \frac{d}{t} - d\left(1 - \left(1 - \frac{1}{t}\right)^{ct} + \xi\right) \\ &= d\left(c - 1 + \left(1 - \frac{1}{t}\right)^{ct} - \xi\right) \\ &\geqslant d\,(\ln 2 - \varepsilon)\,. \end{aligned}$$

  The last inequality holds for t sufficiently large and ξ sufficiently small and, for instance, $c = 2$.

This concludes the proof of the theorem. □

We go on with a further hardness result and consider the decision problem: Given an input matrix A and a set of allowed MLC segments $\mathcal{S}'$, are there nonnegative integers u_S such that $A = \sum_{S \in \mathcal{S}'} u_S S$? If we can prove the NP-hardness of the decision problem, the NP-hardness of **Approx-MIN-TC** immediately follows, because A is decomposable iff **Approx-MIN-TC** yields a TC of 0. Furthermore, this again implies the NP-hardness of the **CVP**.

Theorem 18. *The decision problem described above is NP-hard, if the input matrix A is binary and has two rows.*

Proof. We will prove this by a reduction from the Exact-3SAT-problem [34]. Let us recall the problem:

- **Instance:** A CNF formula in which each clause contains exactly three literals.
- **Question:** Is there an assignment to the variables such that the formula is satisfied?

Let an Exact-3SAT instance Φ in the variables $x_1, \ldots, x_s$ be given. Let $c_1, \ldots, c_t$ denote the clauses of Φ. Let n_i denote the number of clauses that contain the variable x_i and m_i the number of clauses that contain the negation of x_i. We build an instance of the decision problem as follows: A is a matrix with two rows. In the first row of A, we put s consecutive intervals $I(x_i)$ $(i \in [s])$ of ones belonging to the variables. The interval

$I(x_i)$ consists of $2\max(n_i, m_i)-1$ consecutive ones (we assume that each variable occurs at least once). We let also $I(\overline{x}_i) := I(x_i)$. In the second row of A, we similarly put t consecutive intervals $I(c_j)$ ($j \in [t]$) of ones, each containing 5 consecutive ones and corresponding to clause c_j. If one row now has more entries than the other, we fill the other with zeros until both rows have the same number of entries denoted by n. Note that we have s variables and at most $\binom{s}{3}$ clauses. Thus, $n = O(ts^3)$ which is a polynomial size.
In this proof, for the sake of simplicity, we identify intervals of the form $[l, r]$ and the $1 \times n$ vectors they represent. For each interval $I(x_i) = [l, l+2k]$ with an odd number of ones, we consider two decompositions into sub-intervals that correspond to setting the variable x_i true or false. The decomposition corresponding to setting x_i true is $I(x_i) = [l, l]+[l+1, l+2]+[l+3, l+4]+\cdots+[l+2k-1, l+2k]$. The decomposition corresponding to setting x_i false is $I(x_i) = [l, l+1]+[l+2, l+3]+\cdots+[l+2k-2, l+2k-1]+[l+2k, l+2k]$. An illustration for an interval containing 5 ones is given in Figure 6.2.

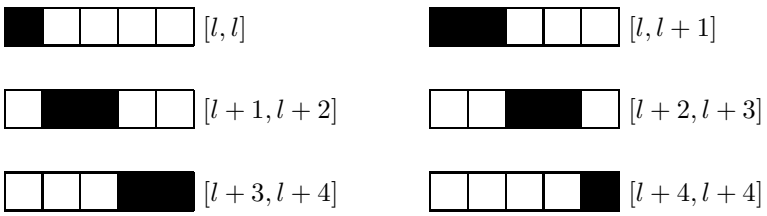

Fig. 6.2: The sub-intervals used for the variables. The decomposition for setting x_i true is on the left and the one for setting x_i false is on the right.

Similarly, to each clause c_j there corresponds an interval $I(c_j) = [5j-4, 5j]$ of 5 consecutive ones in the second row of A. We define ten sub-intervals that can be combined in several ways to decompose $I(c_j)$. We let $I_1(c_j) := [5j-4, 5j-4]$, $I_2(c_j) := [5j-2, 5j-2]$, $I_3(c_j) := [5j, 5j]$, $I_4(c_j) := [5j-3, 5j-3]$, $I_5(c_j) := [5j-1, 5j-1]$, $I_6(c_j) := [5j-4, 5j-3]$, $I_7(c_j) := [5j-1, 5j]$, $I_8(c_j) := [5j-3, 5j-1]$, $I_9(c_j) := [5j-4, 5j-1]$ and $I_{10}(c_j) := [5j-3, 5j]$. An illustration is given in Figure 6.3.

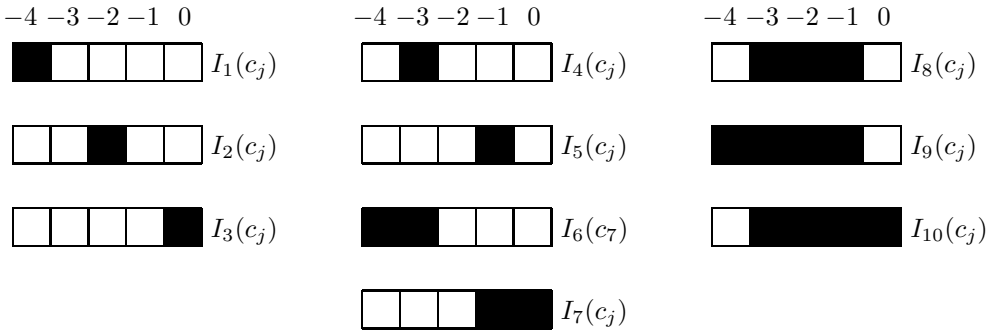

Fig. 6.3: The sub-intervals used for the clauses.

The three first sub-intervals $I_\alpha(c_j)$ ($\alpha \in \{1, 2, 3\}$) correspond to the three literals of the clause c_j. The last seven sub-intervals $I_\alpha(c_j)$ ($\alpha \in \{4, \ldots, 10\}$) alone are not sufficient

to decompose $I(c_j)$ exactly. In fact, if we prescribe any subset of the first three sub-intervals in a decomposition, we can complete the decomposition using some of the last seven intervals to an exact decomposition of $I(c_j)$ in all cases but one: If none of the three first sub-intervals is part of the decomposition, the best we can do is to approximate $I(c_j)$ by, e.g., $I_9(c_j)$, resulting in a total change of 1 for the interval.
We now iteratively define the segments of the instance of the decision problem. We call a sub-interval of an interval $I(x_i)$ (either from the first or the second decomposition) unmatched if it has not yet build a segment with a corresponding second row partner. The first $3t$ segments correspond to pairs (y_i, c_j) where $y_i \in \{x_i, \bar{x}_i\}$ is a literal and c_j is a clause involving y_i. If we have a pair of the form (x_i, c_j) and x_i is the α-th literal of c_j, then we build a segment that irradiates $I_\alpha(c_j)$ in the second row and an unmatched sub-interval from the first decomposition of $I(x_i)$ in the first row. Analogously, if we have a pair of the form $(\overline{x}_i, c_j)$ and $\overline{x}_i$ is the α-th literal of c_j, then we build a segment that irradiates $I_\alpha(c_j)$ in the second row and an unmatched sub-interval from the second decomposition of $I(x_i)$ in the first row. The choice of $2\max(n_i, m_i) - 1$ ones in interval $I(x_i)$ ensures, that sufficiently many unmatched sub-intervals are available. Furthermore, we add all segments with an empty first row and an open sub-interval of the form $I_\gamma(c_j)$ in the second row, where c_j is a clause and $\gamma \in \{4, \ldots, 10\}$. And finally we add all segments with an empty second row and an unmatched remaining sub-interval from the first row. We denote the resulting set of segments by $\mathcal{S}'$. This concludes the description of the reduction. Note that the reduction is clearly polynomial.

Let us now show that the 3SAT-instance Φ is satisfiable iff the corresponding matrix A is decomposable into the set of segments from $\mathcal{S}'$ chosen above. Firstly, let Φ be satisfiable and let an assignment be given that satisfies Φ. For all variables x_i assigned true, we use the segments from the first decomposition and for all variables x_i assigned false, we use the segments from the second decomposition from above. Thus, the first row of A is completely decomposed and for each interval $I(c_j)$ in the second row at least one of the sub-intervals $I_1(c_j)$, $I_2(c_j)$ and $I_3(c_j)$ is irradiated by definition of the segments. Obviously, no one of A is irradiated twice and the remaining ones can be irradiated by choosing appropriate extra segments with closed first row. Thus, A is decomposable. Secondly, let A be decomposable into our given set of segments. Then each interval $I(x_i)$ is decomposed either by the first or by the second decomposition, because if segments from both decompositions are used, it is not possible to cover all ones. Whenever an interval $I(x_i)$ is decomposed by the first decomposition from above, we assign x_i true and whenever the second decomposition from above is used, we assign x_i false. As the second row is correctly decomposed, we know from our previous argumentation that each clause is satisfied and thus Φ is satisfied.
All in all, we have provided a polynomial transformation between an Exact-3SAT-instance and an instance of the decision problem for two rows with a binary input matrix, such that the 3SAT-instance is satisfiable iff the answer to the corresponding decision problem is yes. This yields the NP-hardness of the decision problem for binary matrices with two rows. □

Example 8. For the 3SAT-instance $(x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3)$, we get the matrix

$$A = \left(\begin{array}{ccc|ccc|ccc|ccccc} 1 & 1 & 1| & 1 & 1 & 1| & 1 & 1 & 1| & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1| & 1 & 1 & 1 & 1 & 1| & 1 & 1 & 1 & 1 & 1 \end{array} \right)$$

and the following segments, where $S = ([l, r], [l', r'])$ is the segment, that has ones in $[l, r]$ in the first row, ones in $[l', r']$ in the second row and zeros elsewhere:

1. decomposition $I(x_1)$: $([1,1], I_1(c_1) = [1,1])$, $([2,3], I_1(c_3) = [11,11])$
2. decomposition $I(x_1)$: $([1,2], I_1(c_2) = [6,6])$, $([3,3], \emptyset)$
1. decomposition $I(x_2)$: $([4,4], I_2(c_2) = [8,8])$, $([5,6], I_2(c_3) = [13,13])$
2. decomposition $I(x_2)$: $([4,5], I_2(c_1) = [3,3])$, $([6,6], \emptyset)$
1. decomposition $I(x_3)$: $([7,7], I_3(c_1) = [5,5])$, $([8,9], I_3(c_2) = [10,10])$
2. decomposition $I(x_3)$: $([7,8], I_3(c_3) = [15,15])$, $([9,9], \emptyset)$

further segments $I(c_1)$: $(\emptyset, [2,2])$, $(\emptyset, [4,4])$, $(\emptyset, [1,2])$, $(\emptyset, [4,5])$
$(\emptyset, [2,4])$, $(\emptyset, [1,4])$, $(\emptyset, [2,5])$

further segments $I(c_2)$: $(\emptyset, [7,7])$, $(\emptyset, [9,9])$, $(\emptyset, [6,7])$, $(\emptyset, [9,10])$
$(\emptyset, [7,9])$, $(\emptyset, [6,9])$, $(\emptyset, [7,10])$

further segments $I(c_3)$: $(\emptyset, [12,12])$, $(\emptyset, [14,14])$, $(\emptyset, [11,12])$, $(\emptyset, [14,15])$
$(\emptyset, [12,14])$, $(\emptyset, [11,14])$, $(\emptyset, [12,15])$

and the possible truth assignment $x_1 \wedge x_2 \wedge \overline{x}_3$ corresponds to the segmentation

$$\begin{aligned} A = \quad & 1.\ \text{decomp. } I(x_1) + 1.\ \text{decomp. } I(x_2) + 2.\ \text{decomp. } I(x_3) \\ & + (\emptyset, [2,5]) + (\emptyset, [6,7]) + (\emptyset, [9,10]) + (\emptyset, [12,12]) + (\emptyset, [14,14]). \end{aligned}$$

We state a further hardness result that tightens Theorem 18 for the two row case, that was formulated and proved in [29]. It shows, that **Approx-MIN-TC** is also NP-hard to approximate.

Theorem 19 (Engelbeen, Fiorini)**.** *There exists some $\varepsilon > 0$ such that the* **CVP***, restricted to $2 \times n$ matrices and generators with their ones consecutive on each row, admits no polynomial-time $(\Delta_\infty, \Delta_1)$-approximation algorithm with $\Delta_1 \leqslant \varepsilon\, n$, unless $P = NP$.*

Now we give a polynomial-time $\big(O(\sqrt{d \ln d}\,), O(d\sqrt{d \ln d}\,)\big)$-approximation algorithm for the **CVP**. This algorithm rounds an optimal solution of the LP relaxation of the **CVP** given in Section 6.1.1 (see page 59).
If the LP relaxation (LP) is infeasible, the same holds for corresponding **CVP** instance. Now assume that (LP) is feasible and let LP denote the value of an optimal solution of (LP). Obviously, we have $OPT \geqslant LP$.
Note that for each basic feasible solution of (LP), there are at most d components of $\boldsymbol{u}$ that are nonzero. This is the case, because if we assume that $q > d$ nonzero coefficients exist, then only $k - q$ inequalities of type (6.8) are satisfied with equality. As we have $2d+k$ variables, we need at least $2d+k$ independent equalities to define a vertex. Thus, there must be $(2d + k) - (k - q) - d = d + q > 2d$ independent inequalities of type (6.4), (6.5), (6.6) and (6.7) that are satisfied with equality. This is a contradiction, as there can be at most $2d$ such inequalities. Thus, for any extremal optimal solution of the linear program, at most d of the coefficients u_j are nonzero.

Algorithm 6 Randomized approximation algorithm for the CVP

Input: $\boldsymbol{a} \in \mathbb{Z}_+^d$, $C \in \mathbb{Z}_+ \cup \{\infty\}$, and $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k \in \{0,1\}^d$.

Output: An approximation $\tilde{\boldsymbol{b}}$ of $\boldsymbol{a}$.

If (LP) is infeasible, report that the **CVP** instance is infeasible.

Otherwise, compute an extremal optimal solution $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{u}^*)$ of (LP).

for all $j \in [k]$ **do**

$$\text{if } u_j^* \text{ is integer } \tilde{u}_j := u_j^*, \text{ otherwise } \tilde{u}_j := \begin{cases} \lceil u_j^* \rceil \text{ with probability } u_j^* - \lfloor u_j^* \rfloor, \\ \lfloor u_j^* \rfloor \text{ with probability } \lceil u_j^* \rceil - u_j^*. \end{cases}$$

end for

Return $\tilde{\boldsymbol{b}} := \sum_{j=1}^k \tilde{u}_j \boldsymbol{g}_j$.

Algorithm 6 is an application of the so called randomized rounding technique. This is a widespread technique for approximating combinatorial optimization problems, see e.g. the survey by Motwani, Naor and Raghavan [58]. A basic problem where randomized rounding is used is the *lattice approximation problem*: given a binary matrix H of size $d \times d$ and a rational column vector $\boldsymbol{x} \in [0,1]^d$, find a binary vector $\boldsymbol{y} \in \{0,1\}^d$ that minimizes $\|H(\boldsymbol{x} - \boldsymbol{y})\|_\infty$.

We will use the following result due to Motwani et al., which is a consequence of the Chernoff bound.

Theorem 20 ([58])**.** *Let $(H, \boldsymbol{x})$ be an instance of the lattice approximation problem, and let $\boldsymbol{y}$ be the binary vector obtained by letting $y_j = 1$ with probability x_j and $y_j = 0$ with probability $1 - x_j$, independently, for $j \in [d]$. Then the resulting rounded vector $\boldsymbol{y}$ satisfies $\|H(\boldsymbol{x} - \boldsymbol{y})\|_\infty \leqslant \sqrt{4d \ln d}$ with probability at least $1 - \frac{1}{d}$.*

We resume our discussion of Algorithm 6. By the discussion above, we know that at most d of the components of $\boldsymbol{u}^*$ are nonzero. W.l.o.g. we can assume that all nonzero components of $\boldsymbol{u}^*$ are among its d first components. Then, we let H be the $d \times d$ matrix formed of the first d columns of G. (W.l.o.g. we may assume that $d \leqslant k$. If this is not the case we can add generators consisting only of zeros.) Next, we let $\boldsymbol{x} \in [0,1]^d$ be defined via the following equation (where the floor of the $\boldsymbol{u}^*$ is computed component-wise):

$$\boldsymbol{u}^* - \lfloor \boldsymbol{u}^* \rfloor = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{0} \end{pmatrix}.$$

Finally, the relationship between the rounded vectors is as follows:

$$\tilde{\boldsymbol{u}} - \lfloor \boldsymbol{u}^* \rfloor = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix}.$$

We obtain the following result.

Theorem 21. *Algorithm 6 is a randomized polynomial-time algorithm that either successfully concludes that the given* **CVP** *instance is infeasible, or returns a vector $\tilde{\boldsymbol{b}}$ that is a nonnegative integer linear combination of the generators and satisfies $\|\boldsymbol{a} - \tilde{\boldsymbol{b}}\|_\infty \leqslant C + \sqrt{4d \ln d}$ and $\|\boldsymbol{a} - \tilde{\boldsymbol{b}}\|_1 \leqslant OPT + d\sqrt{4d \ln d}$ with probability at least $1 - \frac{1}{d}$.*

Proof. W.l.o.g. assume that (LP) is feasible. Thus, Algorithm 6 returns an approximation $\tilde{\boldsymbol{b}}$ of $\boldsymbol{a}$. Let $\boldsymbol{b^*} = \sum_{j=1}^{k} u_j^* \boldsymbol{g}_j$. By Theorem 20 and by the discussion above, we have

$$\|\tilde{\boldsymbol{b}} - \boldsymbol{b^*}\|_\infty = \|G\,(\tilde{\boldsymbol{u}} - \boldsymbol{u^*})\,\|_\infty = \|H\,(\boldsymbol{x} - \boldsymbol{y})\,\|_\infty \leqslant \sqrt{4d \ln d},$$

with probability at least $1 - \frac{1}{d}$. Now, the result follows from the inequalities

$$\|\boldsymbol{a} - \tilde{\boldsymbol{b}}\|_\infty \leqslant \|\boldsymbol{a} - \boldsymbol{b^*}\|_\infty + \|\boldsymbol{b^*} - \tilde{\boldsymbol{b}}\|_\infty \leqslant C + \|\boldsymbol{b^*} - \tilde{\boldsymbol{b}}\|_\infty$$

and

$$\|\boldsymbol{a} - \tilde{\boldsymbol{b}}\|_1 \leqslant \|\boldsymbol{a} - \boldsymbol{b^*}\|_1 + \|\boldsymbol{b^*} - \tilde{\boldsymbol{b}}\|_1 \leqslant LP + \|\boldsymbol{b^*} - \tilde{\boldsymbol{b}}\|_1 \leqslant OPT + d\,\|\boldsymbol{b^*} - \tilde{\boldsymbol{b}}\|_\infty.$$

□

By a result of Raghavan [62], Algorithm 6 can be derandomized, at the cost of multiplying the additive approximation guarantees $\sqrt{4d \ln d}$ and $d\sqrt{4d \ln d}$ by a constant. We obtain the following result:

Corollary 22. *There exists a polynomial-time* $\big(O(\sqrt{d \ln d}\,), O(d\sqrt{d \ln d}\,)\big)$*-approximation algorithm for the* **CVP**.

In the case where $C = \infty$, we can slightly improve Theorem 21, as follows:

Theorem 23. *Suppose* $C = \infty$. *Then, Algorithm 6 is a randomized polynomial-time algorithm that returns a vector* $\tilde{\boldsymbol{b}}$ *that is a nonnegative integer linear combination of the generators and satisfies* $\|\boldsymbol{a} - \tilde{\boldsymbol{b}}\|_1 \leqslant OPT + \sqrt{\frac{\ln 2}{2}}\, d\sqrt{d}$ *on average.*

Our proof of Theorem 23 uses the following lemma, which is proved in [29].

Lemma 17. *Let* q *be a positive integer and let* $X_1, X_2, \ldots, X_q$ *be* q *independent random variables such that, for all* $j \in [q]$, $P[X_j = 1 - p_j] = p_j$ *and* $P[X_j = -p_j] = 1 - p_j$. *Then*

$$E\Big[\,|X_1 + X_2 + \cdots + X_q|\,\Big] \leqslant \sqrt{\frac{\ln 2}{2}}\,\sqrt{q}.$$

We are now ready to prove the theorem.

Proof. [Theorem 23] We have

$$\begin{aligned} E\left[\left\|\boldsymbol{a} - \tilde{\boldsymbol{b}}\right\|_1\right] &\leqslant E\Big[\left\|\boldsymbol{a} - \boldsymbol{b^*}\right\|_1\Big] + E\left[\left\|\boldsymbol{b^*} - \tilde{\boldsymbol{b}}\right\|_1\right] \\ &= LP + E\left[\left\|\sum_{j=1}^{k} u_j^* \boldsymbol{g}_j - \sum_{j=1}^{k} \tilde{u}_j \boldsymbol{g}_j\right\|_1\right] \\ &= LP + E\left[\sum_{i=1}^{d}\left|\sum_{j=1}^{k} g_{ij}\,(u_j^* - \tilde{u}_j)\right|\right]. \end{aligned}$$

Without loss of generality, we may assume that $u_j^* = 0$, and thus $\tilde{u}_j = 0$, for $j > d$. This is due to the fact that $\boldsymbol{u}^*$ is a basic feasible solution, see the above discussion.

Now, let $X_{ij} := g_{ij}\left(u_j^* - \tilde{u}_j\right)$ for all $i, j \in [d]$. For each fixed $i \in [d]$, $X_{i1}, \ldots, X_{id}$ are independent random variables satisfying $X_{ij} = 0$ if $g_{ij} = 0$ or $u_j^* \in \mathbb{Z}_+$ and otherwise

$$X_{ij} = \begin{cases} u_j^* - \lceil u_j^* \rceil & \text{with probability } u_j^* - \lfloor u_j^* \rfloor, \\ u_j^* - \lfloor u_j^* \rfloor & \text{with probability } \lceil u_j^* \rceil - u_j^*. \end{cases}$$

By Lemma 17, we get:

$$\begin{aligned} E\left[\left\|\boldsymbol{a} - \tilde{\boldsymbol{b}}\right\|_1\right] &\leqslant LP + E\left[\sum_{i=1}^{d} |X_{i1} + X_{i2} + \cdots + X_{id}|\right] \\ &= LP + \sum_{i=1}^{d} E\Big[\, |X_{i1} + X_{i2} + \cdots + X_{id}| \,\Big] \\ &\leqslant LP + \sqrt{\frac{\ln 2}{2}}\, d\sqrt{d} \\ &\leqslant OPT + \sqrt{\frac{\ln 2}{2}}\, d\sqrt{d}. \end{aligned}$$

□

A natural question is the following: Is it possible to derandomize Algorithm 6 in order to obtain a polynomial-time approximation algorithm for the **CVP** that provides a total change of at most $OPT + O(d\sqrt{d}\,)$, provided that $C = \infty$? We leave this question open.

6.1.3 Some problem generalizations

In this section we generalize our results for the **CVP** to the case where we do not only want to minimize the total change, but a combination of the total change and the delivery time $\sum_{j=1}^{k} u_j$. More precisely, we replace the original objective function $||\boldsymbol{a} - \boldsymbol{b}||_1$ by

$$\mu \cdot ||\boldsymbol{a} - \boldsymbol{b}||_1 + \nu \cdot \sum_{j=1}^{k} u_j,$$

where μ and ν are arbitrary nonnegative importance factors. Throughout this section, we study the **CVP** under this objective function. The resulting problem is denoted by **CVP-DT**.

Here, we observe that the main results of the previous sections still hold with the new objective function. First, for the hardness results, this is obvious because taking $\mu = 1$ and $\nu = 0$ gives back the original objective function. Second, for showing that **CVP-DT** is polynomial when the matrix G defined by the generators is totally unimodular,

we use the following LP relaxation:

$$\text{(LP')} \quad \min \mu \cdot \sum_{i=1}^{d} (\alpha_i + \beta_i) + \nu \cdot \sum_{j=1}^{k} u_j$$

$$\begin{aligned}
\text{subject to} \quad \sum_{j=1}^{k} u_j g_{ij} - \alpha_i + \beta_i &= a_i && \forall i \in [d]\,, \\
\alpha_i &\geqslant 0 && \forall i \in [d]\,, \\
\beta_i &\geqslant 0 && \forall i \in [d]\,, \\
\alpha_i &\leqslant C && \forall i \in [d]\,, \\
\beta_i &\leqslant C && \forall i \in [d]\,, \\
u_j &\geqslant 0 && \forall j \in [k]\,.
\end{aligned}$$

Furthermore, if the columns of G satisfy the consecutive ones property, we can still give a direct reduction to the minimum cost flow problem. Indeed, it suffices to redefine the cost of the arcs of D by letting the cost of arcs of the form $(j, j+1)$ or $(j+1, j)$ (for $j \in [d]$) be μ, and the costs of the other arcs be ν.

Remark 1. It is easy to see that in the minimum cost flow formulation for the one row case of **CVP-DT** the arc capacities C can be generalized if we want to restrict the allowed deviations as in Section 5.1. If we have lower and upper dose bounds $\underline{a}_i \leq b_i \leq \overline{a}_i$ for $i \in [d]$ we can put the capacity $a_j - \underline{a}_j$ to the arcs of type $(j, j+1)$ and the capacity $\overline{a}_j - a_j$ to the arcs of type $(j+1, j)$. By taking $\mu = 1$ and ν large enough, this minimum cost flow formulation obviously provides an alternative algorithm for the problem **Approx-MIN-DT-TC-Row**, although the approach from Section 5.1 outperforms this formulation with respect to computation time. Furthermore, the more precise dose restrictions can also be integrated into the LP-formulation for the general **CVP-DT** that is used for the approximation algorithm.

Finally, we can also find an $\left(O(\sqrt{d \ln d}), O(d\sqrt{d \ln d})\right)$-approximation algorithm for **CVP-DT**, by using an extension of the randomized rounding technique due to Srivinasan [69], and its recent derandomization by Doerr and Wahlström [22]. Consider an instance $(H, \boldsymbol{x})$ of the lattice approximation problem. Assume that $\sum_{j=1}^{d} x_j \in \mathbb{Z}_+$. We wish to round $\boldsymbol{x}$ to a binary vector $\boldsymbol{y}$ such that $\sum_{j=1}^{d} x_j = \sum_{j=1}^{d} y_j$ and $\|H(\boldsymbol{x}-\boldsymbol{y})\|_\infty = O(\sqrt{d \ln d})$. Srivinasan [69] obtained a randomized polynomial-time algorithm achieving this with high probability. A recent result of Doerr and Wahlström implies the following theorem.

Theorem 24 ([22])**.** *Let $H \in \{0,1\}^{d \times d}$ and $\boldsymbol{x} \in \mathbb{Q}^d \cap [0,1]^d$ such that $\sum_{j=1}^{d} x_j \in \mathbb{Z}_+$. Then, a binary vector $\boldsymbol{y}$ can be computed in time $O(d^2)$ such that $\sum_{j=1}^{d} y_j = \sum_{j=1}^{d} x_j$ and*

$$\|H(\boldsymbol{x} - \boldsymbol{y})\|_\infty \leqslant (e-1)\sqrt{d \ln d}.$$

Let again $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{u}^*)$ denote any extremal optimal solution of (LP'). Recall that at most d of the k components of $\boldsymbol{u}^*$ are nonzero. Without loss of generality, we can assume that $u_j^* = 0$ for $j > d$.
Now, define H and $\boldsymbol{x}$ as previously. Because it might happen that $\sum_{j=1}^{d} x_j \notin \mathbb{Z}_+$, we turn H and $\boldsymbol{x}$ respectively into a $(d+1) \times (d+1)$ matrix and a $(d+1) \times 1$ vector by letting $x_{d+1} := \left\lceil \sum_{j=1}^{d} x_j \right\rceil - \sum_{j=1}^{d} x_j$ and $h_{d+1,j} = h_{i,d+1} := 0$ for all $i, j \in [d+1]$.
By Theorem 24, one can find in $O(d^2)$ time a vector $\boldsymbol{y} \in \{0,1\}^{d+1}$ such that $\sum_{j=1}^{d+1} y_j = \sum_{j=1}^{d+1} x_j$ and $\|H(\boldsymbol{x} - \boldsymbol{y})\|_\infty \leqslant (e-1)\sqrt{(d+1)\ln(d+1)} = O(\sqrt{d \ln d})$. We then let $\tilde{u}_j = \lfloor u_j^* \rfloor + y_j$ for $j \in [d]$ and $\tilde{u}_j = 0$ for $j \in [k] \backslash [d]$. The corresponding approximation of $\boldsymbol{a}$ is $\tilde{\boldsymbol{b}} := G\tilde{\boldsymbol{u}}$. Notice that the delivery time will be rounded to $\lfloor \sum_{j=1}^{d} u_j^* \rfloor$ if $y_{d+1} = 1$ and to $\lceil \sum_{j=1}^{d} u_j^* \rceil$ if $y_{d+1} = 0$. Using similar arguments as those used in the proof of Theorem 21, we see that Theorem 24 leads to a polynomial-time $\left(O(\sqrt{d \ln d}), O(d\sqrt{d \ln d})\right)$-approximation algorithm for **CVP-DT**.

6.2 Approximation with leaf overtravel constraint

Some MLCs are not capable of shifting the left (respectively right) leaves further to the right (respectively left) than up to a given threshold. That means we are given two parameters $b_\ell, b_r \in [n]$ with $b_\ell \geq b_r + 1$ and require $\ell_i \leq b_\ell$ and $r_i \geq b_r$ for all $i \in [m]$ (cf. definition in Section 3). For example, the electron MLC at the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf can shift the leaf edges to 3/4 of the radiation field.
We therefore give a solution of **Approx-MIN-TC** for $\mathcal{S}' = \mathcal{S}_{LOC}$ with parameters b_ℓ and b_r for the matrix A. As the leaf overtravel constraint only affects a single row of the matrix, the problem can be solved for each row independently. Thus, we compute an optimal approximation of a vector $\boldsymbol{a}$. Segmentations reduce to sums of intervals $[\ell, r]$. Segments are simply binary vectors $\boldsymbol{s}$ with consecutive ones.

Lemma 18. *A vector $\boldsymbol{a}$ has a segmentation $\boldsymbol{a} = \sum_{i=1}^{k} \boldsymbol{s}_i$ with corresponding leaf positions $l_i \leq b_\ell$ and $r_i \geq b_r$ iff $a_j \geq a_{j+1}$ for all $j \in [b_\ell, n-1]$ and $a_j \geq a_{j-1}$ for all $j \in [2, b_r]$.*

Proof. Let $a_0 = a_{n+1} := 0$. On the one hand, the algorithm of Bortfeld (cf. [11]) provides a segmentation where the left leaf position is j for exactly $(a_j - a_{j-1})_+$ segments. Analogously, the right leaf position is j for $(a_j - a_{j+1})_+$ segments and no other leaf positions occur. On the other hand, it is obvious that if $a_j > a_{j-1}$ (respectively $a_j > a_{j+1}$) there will be a segment with left (respectively right) leaf position j in every segmentation. This concludes the proof. □

Therefore, we have to find an approximation vector, that has no up-steps after index b_l and no down-steps before index b_r. As we assume $b_r < b_\ell$, we can use symmetry to solve the approximation problem for the right leaf positions. Besides, the criterion from Lemma 18 shows, that $b_j = a_j$ for $j \in [b_r + 1, b_\ell - 1]$ for each optimal solution of the problem. We simply need to solve the following problem for the subvector $(a_{b_\ell}, \ldots, a_n)$:

LOC-left: Given a vector $\boldsymbol{v} = (v_1, \ldots, v_k)$, find an approximation vector $\boldsymbol{w}$ with $w_j \geq w_{j+1}$ for $j \in [k-1]$ such that $\| \boldsymbol{v} - \boldsymbol{w} \|_1 = \sum_{j=1}^{k} |v_j - w_j| \to \min$.

The algorithm for solving the problem **LOC-left** is described in Algorithm 11 in the appendix. It uses a graph theoretical approach and computes a shortest path in a layered digraph, where the j-th layer consists of nodes representing the possible entries of the j-th component of the approximation vector. The problem **LOC-left** is similar to the **Monotone Discrete Approximation Problem (MDAP)** formulated in [26] and the algorithm follows the same idea.

Let $min := \min_{j \in [k]} v_j$ and $max := \max_{j \in [k]} v_j$ and let tc_{ij} be the objective value of an optimal solution of **LOC-left** for $(v_1, \ldots, v_j)$ with $w_j = i$. Let pre_{ij} be the corresponding predecessor w_{j-1}. With respect to Algorithm 11 (that uses the notation from above) we have the following

Theorem 25. *Algorithm 11 computes an optimal solution of* ***LOC-left****.*

Proof. The initial values tc_{i1} are trivially correct. Let now $j > 1$ and let $(w_1, \ldots, w_j)$ be an optimal approximation of $(v_1, \ldots, v_j)$ with $w_j = i$. By induction, $tc_{w_{j-1},j-1}$ is computed correctly and thus

$$\sum_{\ell=1}^{j} |v_\ell - w_\ell| = tc_{w_{j-1},j-1} + |v_j - i| \geq tc_{ij}.$$

Therefore tc_{ij} is a lower bound for the total change. The choice of i_{opt} makes sure, that the optimal value of w_j is chosen and obviously the approximation vector from Algorithm 11 realizes the lower bound for the total change of $tc_{i_{opt},k}$. □

6.3 A column generation approach to total change minimization

We have recognized in Section 6.1 that the approximate segmentation problems aiming at minimizing the total change are NP-hard in general. The approximation algorithm described there can only be used if the set of available segments $\mathcal{S}'$ is explicitly given. But if $\mathcal{S}'$ is e.g. given as the set of segments satisfying some specified constraint, there can still be exponentially many segments in $\mathcal{S}'$ and there is no chance of solving the LP relaxation of the **CVP**. For these cases, we need other algorithms to find good solutions for **Approx-MIN-TC**. We therefore introduce a column generation approach to this problem, which is a method that is well known and well-studied for different kinds of applications (cf. [14, 15, 63] for column generation in IMRT). The method has slight disadvantages, e.g. large computation times.

Let, as always, our set of feasible segments be denoted by $\mathcal{S}'$ (maybe implicitly given by a number of constraints) and let (more general than in Section 6.1) $\| \cdot \|$ be a specified vector norm that measures the deviation between the intensity matrix and

the approximation matrix. Recall that the problem **Approx-MIN-TC** is

$$\begin{aligned} \min \left\| A - \sum_{S \in \mathcal{S}'} u_S S \right\| \quad & \text{subject to} \\ u_S \geq 0 \quad & \forall S \in \mathcal{S}', \\ u_S \in \mathbb{Z} \quad & \forall S \in \mathcal{S}'. \end{aligned}$$

The basic idea is to iteratively solve two problems: the problem **Approx-MIN-TC** using only a small explicitly given subset $\mathcal{S}'' \subseteq \mathcal{S}'$ of the allowed segments (master problem) and then compute a segment $S \in \mathcal{S}'$ that might improve the objective function (subproblem) and that is added to $\mathcal{S}''$. We stop if no segment that improves the objective function can be found. We consider the problem **Approx-MIN-TC** with the ℓ_1-norm and with the ℓ_2-norm as measure for the deviation.
In the literature, column generation approaches are often used for problems without integrality constraints. The Karush-Kuhn-Tucker conditions for optimality for continuous optimization problems can be used. Therefore, at first we discuss the column generation method for the relaxed problem version and then give remarks for our algorithmic approach to ensure integral solutions. As the subproblems have the same structure for both norms, we can treat them in the same way. There exist also branch-and-price strategies to solve huge integer problems by column generation methods, see e.g. [7] for a survey.

6.3.1 Approx-MIN-TC with ℓ_1-norm

The relaxed problem **Approx-MIN-TC** where the deviation is measured by the ℓ_1-norm (TC-1) reduces to the linear program

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} (\alpha_{ij} + \beta_{ij}) \text{ subject to} \tag{6.9}$$

$$\sum_{S \in \mathcal{S}'} u_S S_{ij} - \alpha_{ij} + \beta_{ij} = a_{ij} \qquad \forall (i,j) \in [m] \times [n], \quad (w_{ij}) \tag{6.10}$$

$$\alpha_{ij}, \beta_{ij} \geq 0 \qquad \forall (i,j) \in [m] \times [n], \tag{6.11}$$

$$u_S \geq 0 \qquad \forall S \in \mathcal{S}', \tag{6.12}$$

where the variables in parentheses denote the corresponding variables of the dual problem.
The dual TC-1-DUAL of the problem above is

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} a_{ij} \text{ subject to} \tag{6.13}$$

$$w_{ij} \leq 1 \qquad \forall (i,j) \in [m] \times [n], \quad (\alpha_{ij}) \tag{6.14}$$

$$w_{ij} \geq -1 \qquad \forall (i,j) \in [m] \times [n], \quad (\beta_{ij}) \tag{6.15}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} S_{ij} w_{ij} \geq 0 \qquad \forall S \in \mathcal{S}'. \quad (u_S) \tag{6.16}$$

A feasible solution $(\alpha_{ij}, \beta_{ij}, u_S)$ of TC-1 and a feasible solution (w_{ij}) of TC-1-DUAL are optimal, if the complementary slackness conditions for optimality hold:

$$\alpha_{ij}(1 - w_{ij}) = 0 \qquad \forall (i,j) \in [m] \times [n], \tag{6.17}$$

$$\beta_{ij}(w_{ij} + 1) = 0 \qquad \forall (i,j) \in [m] \times [n], \tag{6.18}$$

$$u_S\big(\sum_{i=1}^{m}\sum_{j=1}^{n} S_{ij}w_{ij}\big) = 0 \qquad \forall S \in \mathcal{S}'. \tag{6.19}$$

As there can be a large number of allowed segments or the set of segments is given only implicitly, it is often not possible to solve the problem TC-1 for $\mathcal{S}'$. Therefore, a column generation approach is used, where we alternately solve TC-1 for a current subset $\mathcal{S}'' \subseteq \mathcal{S}'$ and a subproblem that decides for a new segment S such that solving TC-1 for $\mathcal{S}'' \cup \{S\}$ improves the value of the objective function of TC-1.
Let TC-1$_{\mathcal{S}''}$ denote the problem TC-1 where the set of allowed segments is $\mathcal{S}''$ and $\mathcal{S}'' \subseteq \mathcal{S}'$ (and similarly for TC-1-DUAL). We assume, we have computed an optimal solution $(\alpha_{ij}, \beta_{ij}, u_S)$ of TC-1$_{\mathcal{S}''}$ and (w_{ij}) of TC-1-DUAL$_{\mathcal{S}''}$ such that the slackness conditions hold. We set $u_S = 0$ for all $S \in \mathcal{S}' \setminus \mathcal{S}''$. Obviously, we get a feasible solution of TC-1$_{\mathcal{S}'}$ and also the slackness conditions (6.17)-(6.19) are satisfied for $\mathcal{S}'$. As (6.14) and (6.15) also hold, the only inequality that might not be satisfied for a segment $S \in \mathcal{S}' \setminus \mathcal{S}''$ is (6.16). As dual feasibility is necessary for optimality, the subproblem consists in finding a segment $S \in \mathcal{S}' \setminus \mathcal{S}''$ that violates (6.16) as much as possible, i.e. that minimizes

$$\sum_{i=1}^{m}\sum_{j=1}^{n} S_{ij}w_{ij}.$$

Thus, the subproblem has the following form: For each entry (i,j) of the irradiation field $[m] \times [n]$, we have a weight $w_{ij} \in [-1, 1]$. We are looking for a segment $S \in \mathcal{S}'$ that minimizes the sum of the weights of the open bixels.
If solving the subproblem yields a segment where the objective function of the subproblem is negative, we add the segment to $\mathcal{S}''$ and solve TC-1$_{\mathcal{S}''}$ again. If no such segment exists, the current solution of TC-1 is optimal.

Remark 2 (Algorithmic realization)**.** For our discrete segmentation problems, we are actually interested in integer solutions of TC-1. First, we follow the approach for the relaxed problem from above and each time we deal with the master problem we solve the linear program TC-1-DUAL using Gurobi [59] to get the (real-valued) weights for the subproblem. We iterate until no more segment with negative weight can be found and end up with a set of chosen segments $\mathcal{S}'' \subseteq \mathcal{S}'$. Afterwards, we solve the master problem TC-1$_{\mathcal{S}''}$ as an integer linear program using Gurobi [59] again to get our final approximate segmentation with integral coefficients. Thus, we solve a sequence of linear programs and one integer linear program. It turns out that although the choice of segments is done with regard to the relaxed problems, good integral results can be produced. The reason for this is, that experimental tests show that the integrality gap between the optimal solution of the LP TC-1-DUAL and the optimal solution of the ILP TC-1 is mostly zero and always very small. Thus, there is an integer solution whose

value of the objective function is nearly the optimal value of the objective function of the relaxed problem. Of course, other approaches using randomized rounding as in Section 6.1.2 are also applicable for solving the master problem TC-$1_{\mathcal{S}''}$ at the end.

Remark 3 (Generalization)**.** The column generation approach for **Approx-MIN-TC** can be modified to solve the more general **CVP** described in Section 6.1 by adding the constraints $\alpha_{ij} \leq C$ and $\beta_{ij} \leq C$ in TC-1 and changing the dual accordingly. The subproblem does not change. Thus, for the problem with ℓ_1-norm, it is easy to integrate maximum allowed deviations into the column generation model.

6.3.2 Approx-MIN-TC with ℓ_2-norm

The relaxed problem where the deviation is measured by the ℓ_2-norm (TC-2) is

$$\min f(\boldsymbol{u}) = \sum_{i=1}^{m}\sum_{j=1}^{n}\big(a_{ij} - \sum_{S\in\mathcal{S}'} u_S S_{ij}\big)^2 \text{ subject to} \tag{6.20}$$

$$u_S \geq 0 \qquad \forall S \in \mathcal{S}'. \tag{6.21}$$

Note that this problem definition corresponds to minimizing the square of the ℓ_2-norm which is equivalent to minimizing the ℓ_2-norm itself. The objective function can be written as follows:

$$\begin{aligned} f(\boldsymbol{u}) = \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2 - \sum_{i=1}^{m}\sum_{j=1}^{n}\big(2a_{ij}\sum_{S\in\mathcal{S}'} u_S S_{ij}\big) \\ & + \sum_{i=1}^{m}\sum_{j=1}^{n}\big((\sum_{S\in\mathcal{S}'} u_S S_{ij})(\sum_{S'\in\mathcal{S}'} u_{S'} S'_{ij})\big). \end{aligned} \tag{6.22}$$

We define the vector $\boldsymbol{c} = (c_S)_{S\in\mathcal{S}'}$ by $c_S := -2\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}S_{ij}$ and the matrix $D = (d_{S,S'})_{S,S'\in\mathcal{S}'}$ by $d_{S,S'} := 2\sum_{i=1}^{m}\sum_{j=1}^{n} S_{ij}S'_{ij}$. As $\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$ is a constant, the minimization problem is equivalent to

$$\min h(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T D\boldsymbol{u} + \boldsymbol{c}^T\boldsymbol{u} \text{ subject to} \tag{6.23}$$

$$\boldsymbol{u} \geq \boldsymbol{0}. \tag{6.24}$$

This is a quadratic programming problem with nonnegativity constraints. As the entries of D are doubles of inner products of the segments, D is a multiple of a Gram matrix and therefore symmetric and positive semidefinite. Thus, the objective function $h(\boldsymbol{u})$ is convex and obviously $-\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$ is a lower bound for h which ensures the existence of a global minimum. Due to the nonnegativity constraints, however, there does not exist an analytical solution for the global minimum. There exist several methods to solve such problems like the active set methods, interior point methods or iterative approaches as Quasi-Newton or Projected-Newton methods (cf. [67] for details).

The Karush-Kuhn-Tucker (KKT) conditions for optimality are the existence of a vector $\boldsymbol{\lambda}$ such that

$$D\boldsymbol{u} + \boldsymbol{c} = \boldsymbol{\lambda} \tag{6.25}$$
$$\boldsymbol{u} \geq \mathbf{0} \tag{6.26}$$
$$\boldsymbol{\lambda} \geq \mathbf{0} \tag{6.27}$$
$$u_S \lambda_S = 0 \qquad \forall S \in \mathcal{S}'. \tag{6.28}$$

Again, we denote by TC-$2_{\mathcal{S}''}$ the problem TC-2 where the set of allowed segments is $\mathcal{S}''$ and $\mathcal{S}'' \subseteq \mathcal{S}'$. Assume, we have an optimal solution $\boldsymbol{u}$ of TC-$2_{\mathcal{S}''}$ such that the KKT conditions hold. As before, we put $u_S = 0$ for all $S \in \mathcal{S}' \setminus \mathcal{S}''$. This yields a feasible solution of TC-$2_{\mathcal{S}'}$ and also the KKT conditions (6.26) and (6.28) hold. But there might be segments $S \in \mathcal{S}' \setminus \mathcal{S}''$ such that the corresponding row in $D\boldsymbol{u} + \boldsymbol{c} \geq \mathbf{0}$ is violated. Let us therefore illustrate the matrix D and the vector c such that the entries belonging to $\mathcal{S}''$ are the first ones:

$$D = \left(\begin{array}{c|c} D_{\mathcal{S}''} & D'^T \\ \hline D' & D_{\mathcal{S}' \setminus \mathcal{S}''} \end{array}\right) \text{ and } \boldsymbol{c} = \begin{pmatrix} \boldsymbol{c}_{\mathcal{S}''} \\ \hline \boldsymbol{c}' \end{pmatrix}.$$

For the current solution $\boldsymbol{u} = (\boldsymbol{u}_{\mathcal{S}''}, \mathbf{0})^T$, we have

$$\left(\begin{array}{c|c} D_{\mathcal{S}''} & D'^T \\ \hline D' & D_{\mathcal{S}' \setminus \mathcal{S}''} \end{array}\right) \begin{pmatrix} \boldsymbol{u}_{\mathcal{S}''} \\ \hline \mathbf{0} \end{pmatrix} + \begin{pmatrix} \boldsymbol{c}_{\mathcal{S}''} \\ \hline \boldsymbol{c}' \end{pmatrix} = \begin{pmatrix} D_{\mathcal{S}''}\boldsymbol{u}_{\mathcal{S}''} + \boldsymbol{c}_{\mathcal{S}''} \\ \hline D'\boldsymbol{u}_{\mathcal{S}''} + \boldsymbol{c}' \end{pmatrix}.$$

We already know from the optimality of $\boldsymbol{u}_{\mathcal{S}''}$ for the restricted problem TC-$2_{\mathcal{S}''}$, that $D\boldsymbol{u}_{\mathcal{S}''} + \boldsymbol{c}_{\mathcal{S}''} \geq \mathbf{0}$ and a solution is only optimal for the complete problem if for all segments $S \in \mathcal{S}' \setminus \mathcal{S}''$

$$\sum_{S' \in \mathcal{S}''} d_{S,S'} u_{S'} + c_S \geq 0$$

or equivalently, if

$$\min_{S \in \mathcal{S}'} \sum_{S' \in \mathcal{S}''} d_{S,S'} u_{S'} + c_S \geq 0$$

holds. Inserting the definition of the $d_{S,S'}$ and the c_S and regrouping the terms, we get the problem to find a segment $S \in \mathcal{S}' \setminus \mathcal{S}''$ that minimizes

$$2 \sum_{i=1}^{m} \sum_{j=1}^{n} \Big(\sum_{S' \in \mathcal{S}''} S'_{ij} u_{S'} - a_{ij} \Big) S_{ij}.$$

This subproblem has a similar structure as the one in the previous section. We somehow have a weight $w_{ij} = \sum_{S' \in \mathcal{S}''} S'_{ij} u_{S''} - a_{ij}$ for all $(i,j) \in [m] \times [n]$ and want to find a new segment that minimizes the sum of the weights over the open bixels. Therefore, we can use the same approaches for the subproblem as for the problem with the ℓ_1-norm (cf. next subsection).

If no $S \in \mathcal{S}' \setminus \mathcal{S}''$ with negative value of the term above exists, the current solution is optimal for the complete problem. Otherwise, we add the optimal S to $\mathcal{S}''$ and solve TC-$2_{\mathcal{S}''}$ again.

Remark 4 (Algorithmic realization)**.** As solving integer quadratic problems is quite difficult, we cannot easily compute an optimal integral solution of the problem TC-$2_{\mathcal{S}''}$. We therefore use another approach and compute the column generation procedure for the relaxed problem several times. Each time, we take some segments whose coefficients are close to an integer value and start the column generation with the residual matrix again (cf. Algorithm 7). Obviously, this algorithm is very time consuming as each column generation step can already need a large number of operations.

Algorithm 7 Algorithmic realization for column generation with ℓ_2-norm

Input: A, ε
  $B := \mathbf{0}$
  while not finished **do**
    Compute column generation for $A - B$ and get $\mathcal{S}''$, u_S for $S \in \mathcal{S}''$.
    for all $S \in \mathcal{S}''$ **do**
      $\tilde{u}_S$:=the integral value that is closest to u_S
      if $|u_S - \tilde{u}_S| < \varepsilon$ and $\|A - (B + S)\|_2 < \|A - B\|_2$ **then**
        $B := B + S$
        Add S with coefficient $\tilde{u}_S$ to the segmentation.
      end if
    end for
    If no segment was chosen in the for-loop try with a larger ε.
    If no epsilon is found such that we find S with $\|A - (B + S)\|_2 < \|A - B\|_2$, then finish.
  end while
Output: Segmentation, B

Remark 5 (Generalization)**.** It is again possible to incorporate maximum allowed deviations as in the **CVP** by adding the constraints $-C \leq a_{ij} - \sum_{S \in \mathcal{S}'} u_S S_{ij} \leq C$ to the problem. But this yields a quadratic programming problem, that has not only nonnegativity constraints, but also other linear ones. More general algorithms for solving this are needed.

6.3.3 Solving the subproblem

Given the weights w_{ij} for $(i, j) \in [m] \times [n]$, the subproblem amounts to finding a segment $S \in \mathcal{S}'$ that minimizes $\sum_{i=1}^{m} \sum_{j=1}^{n} S_{ij} w_{ij}$. This is a nontrivial problem and its solution, of course, depends on the structure of $\mathcal{S}'$. As this is of special interest in this thesis, we explain the algorithmic approach for the subproblem for $\mathcal{S}' = \mathcal{S}_L$, i.e. for finding large segments according to the definition in Section 3. We are given five parameters $b_\ell, b_r, w \in [n]$, $h, f \in [m]$ with $b_\ell \geq b_r + 1$ and $h \leq f$. The parameter h defines the minimum field size in vertical direction. We distinguish between the cases $h = 1$ and $h > 1$, as the subproblem can be solved optimally for the first case, but not for the second case. These approaches are given now and referred to again in the following Section 6.4 where we deal with these special segment classes in order to find an appropriate segmentation model for clinical practice.

Subproblem for L-segments with $h = 1$

Recall, that we have to compute a connected segment satisfying the leaf overtravel constraint where in each open row at least w bixels are open. The overlap between adjacent open rows also has to be at least w bixels long and at least f consecutive rows have to be open. Let $d_{ii,i,\ell,r}$ denote the cumulated optimal weight up to row i for a segment that has its first nonzero row in row ii and has leaf positions $[\ell, r]$ in row i. Let $pre_{ii,i,\ell,r} = [\ell', r']$ denote the corresponding predecessor leaf positions in row $i-1$. We say that the segment is opened in row ii. Obviously, we need $i, ii \in [m]$, $ii \leq m - f + 1$, $i \geq ii$, $\ell, r \in [n]$, $\ell \leq b_\ell$, $r \geq b_r$ and $\max(\ell, \ell') - \min(r, r') \geq w - 1$ or $[\ell, r] = \emptyset$. Note that $[\ell, r] = \emptyset$ is only allowed if $i > ii + f - 1$, as at least f rows have to be opened. If all these relations between the indices are satisfied, we call (ii, i, ℓ, r) a feasible point and $[\ell', r']$ a feasible predecessor for (ii, i, ℓ, r). Recall that constraint (ii) from the definition of an MFC-segment in Section 3 is trivially satisfied for $h = 1$, as we artificially enlarged the segments to the columns $[-w + 1, n + w]$. Let $weight_{i\ell r} := \sum_{j=\ell}^{r} w_{ij}$.
Obviously, the following facts hold:

- $d_{ii,ii,\ell,r} = weight_{ii,\ell,r}$ for all $ii \leq m - f + 1$ and suitable values of ℓ and r.
- For all feasible points (ii, i, ℓ, r) with $i > ii$, we have

$$d_{ii,i,\ell,r} = \min \left\{ d_{ii,i-1,\ell',r'} + weight_{i\ell r} \;\middle|\; \begin{smallmatrix} (ii,i-1,\ell',r') \text{ feasible point} \\ [\ell',r'] \text{ feasible predecessor} \end{smallmatrix} \right\}$$

  and $pre_{ii,i,\ell,r}$ is an interval $[\ell', r']$ where the minimum is attained.

- The weight of an optimal segment is

$$OPT = \min \left\{ d_{ii,m,\ell,r} \;\middle|\; (ii, m, \ell, r) \text{ feasible point} \right\}.$$

  and the corresponding optimal segment can be found by going backwards using the predecessor information.

This approach can be formulated as a shortest path problem in a directed layered acyclic digraph with feasible points as nodes, arcs between feasible points and their feasible predecessor points and appropriate arc weights according to the given weight function.

Corollary 26. *The procedure described above computes the L-segment for $h = 1$ with minimal weight in $O(m^2 n^4)$ time.*

Subproblem for L-segments with $h > 1$

Now we have to compute a connected segment, whose ones and zeros can be covered by rectangles of size $h \times w$ such that again the leaf overtravel constraint and the overlapping condition between adjacent open rows are satisfied and at least f consecutive rows have to be opened. Obviously, the ones can be covered by $h \times w$-rectangles if they can be covered by rectangles of size h and width of at least w. Thus, to define a segment, we

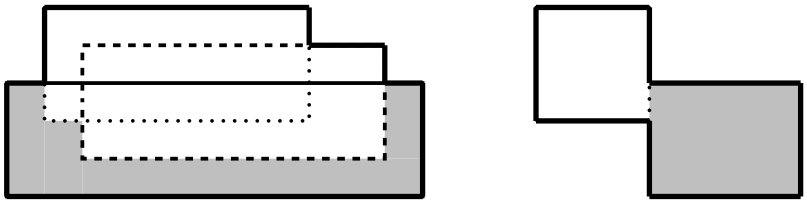

Fig. 6.4: L-segments that can be covered by rectangles of height h and width at least w are determined. The light grey bixels are the new open bixels caused by the choice for the third row.

decide to open a rectangle $[i, i+h-1] \times [\ell, r]$ in each row $i \in [1, m-h+1]$ of the matrix. Note that an empty rectangle with $[\ell, r] = \emptyset$ is possible. Finally, the segment has ones in all bixels belonging to the union of the chosen rectangles.

We cannot expect to compute an optimal segment in a similar way as for the case $h = 1$ here, as for the feasibility of opening a rectangle in row i, we have to consider all the opened rectangles in rows $i-h, \ldots, i-1$. Minimizing over all possible predecessor families in these rows would result in an exponential complexity of $O(n^{2h})$ for the computation of the optimal weight of each node in the shortest path problem (if h is not considered to be a constant and may rise linearly in m). Thus, we are satisfied with a good heuristic solution.

We use the same notation as in the previous section, but this time we want to associate with the choice of the leaf positions $[\ell, r]$ in row i, that we open all bixels of the segment in the rectangle $[i, i+h-1] \times [\ell, r]$. Thus, $d_{ii,i,\ell,r}$ is the cumulated optimal weight of all open bixels from the top to this rectangle if the segment is opened in row ii. Again, $pre_{ii,i,\ell,r}$ is the corresponding predecessor $[\ell', r']$ corresponding to the rectangle $[i-1, \ldots, i+h-2] \times [\ell', r']$. Generalizing the old definition, $weight_{ii,i,\ell,r}$ is the weight that is caused by the new open bixels and that is added to the weight of the predecessor if we choose $[\ell, r]$ in row i (cf. light grey bixels in Figure 6.4). In our algorithm, we proceed as before and minimize over all possible predecessors $[\ell', r']$ in row $i-1$. To compute the new open bixels, we have to follow the predecessor path of $[\ell', r']$ in row $i-1$ up to row $i-h+1$ as this is the first row that causes open bixels in row i. It is important to understand, that we do not minimize over all possible predecessor environments in rows $i-h+1, \ldots, i-1$, but only over all predecessors in row $i-1$ and take their predecessors as given. Thus, our solution is suboptimal and heuristic, but the results are quite satisfying and the computation time becomes acceptable. To save some operations, we do not really go back the predecessor paths in our implementation, but save for each point (ii, i, l, r) the current segment corresponding to the optimal path until row i. Then it is easy to find the new open bixels when opening the next row.

As in the previous section, we have some requirements for the indices: $i, ii \in [m]$, $ii \leq m-f+1$, $ii \leq i \leq m-h+1$, $\ell, r \in [n]$, $\ell \leq b_\ell$, $r \geq b_r$ and $r - \ell \geq w-1$ or $[\ell, r] = \emptyset$. Furthermore, $[\ell', r']$ can only be taken as predecessor for a point (ii, i, ℓ, r), if the following conditions hold:

- The resulting segment is connected, e.g. we may not take $[\ell, r] \neq \emptyset$ if row $i-1$ is totally closed.

- The segment satisfies the overlapping condition, i.e. if there are already open bixels in row i, their union with $[\ell, r]$ must satisfy the consecutive ones property and if there are no open bixels in row i yet, $[\ell, r]$ must overlap with at least w open bixels of row $i-1$.
- Choosing $[\ell, r]$ in row i must not lead to too small regions of zeros in the segment (i.e. regions of height smaller than h).
- If there are no open bixels in row i yet, we may only choose $[\ell, r] = \emptyset$ if at least f rows are already opened.

To check these conditions, we again have to follow the (given) predecessor path of the point $(ii, i-1, \ell', r')$, but now up to row $i-h$ (because of the third condition from above). Again, we use the saved current segment for this. Note that choosing $[\ell, r] = \emptyset$ in some row i does not necessarily mean that the open region of the segment ends here, because the rectangles span several rows and there can already be open bixels in row i. Therefore, open rectangles in the following rows are possible. Taking care of the conditions above, feasible points and feasible predecessors are then defined as in the previous section. Solving the subproblem heuristically again amounts to a shortest path computation. The result of the subproblem is an L-segment with heuristically minimized weight that serves as new candidate for the master problem. We will refer to the column generation approach with subproblem for L-segments in the next section and compare it to other algorithms.

6.4 Approximation with minimum field size constraint - A clinical segmentation model

In this section, we deal with the problem **Approx-MIN-TC** in the case $\mathcal{S}' = \mathcal{S}_L$ (as defined in Chapter 3 at the very end). The model for these classes of segments was built up in collaboration with Tobias Gauer from the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf and, in his mind, it fits to the needs of clinical practice. He developed a treatment planning system for electron irradiation and the following algorithms are integrated into the optimization programme (cf. [25]). The results of this section are based on [52].
On the one hand, the MLCs that are used in Hamburg have the leaf overtravel constraint. On the other hand, segments should satisfy some minimum field size constraints, because the assumption that a treatment plan is optimal, if the linear combination of the chosen segments equals the matrix, does not hold in practice for dosimetric reasons. Such a plan consists of various segments possibly including those segments where almost the whole irradiation field is covered and only few bixels receive radiation. Indeed, the reasons for using larger segments are:

- Irradiation of small photon or electron segments results in a much lower dose output compared to conventional conformal fields. Therefore, the linearity assumption in the discrete segmentation model, that irradiating one segment is equivalent to dividing it into two parts and irradiating them separately, only holds, if the two parts are still sufficiently large.

- The penetration depth of electrons decreases with decreasing field size and is almost independent of the beam energy for approximately 1 cm × 1 cm electron fields. However, the energy dependence of the penetration depth is necessary for our new IMRT technique with electron beams to adjust the dose to the target volume by use of various beam energies. Figure 6.5 shows that electron fields of at least 3 cm × 3 cm are necessary to keep an output factor of nearly 1 and an energy-dependent penetration depth.

- Larger segments are much less liable to negative effects due to breathing motion of the patient.

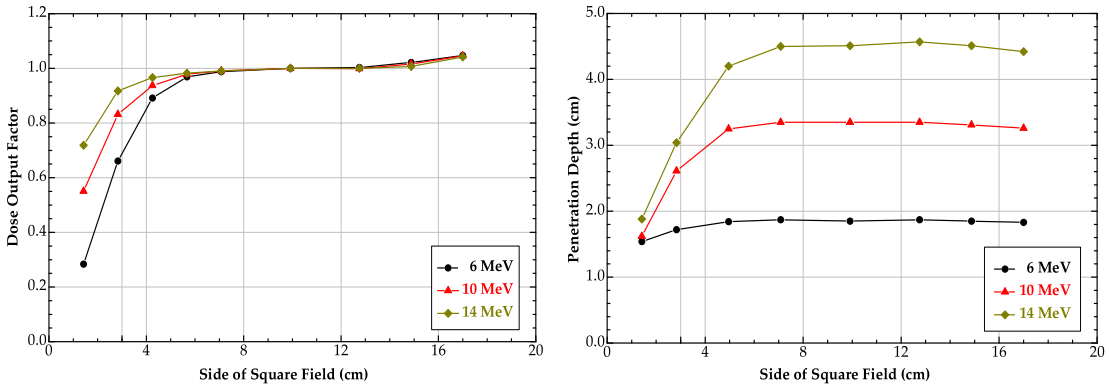

Fig. 6.5: Electron dose output at the dose maximum normalized to the dose output of the 10 cm × 10 cm field and electron penetration depth of the 90 % depth-dose as a function of square field size and electron energy (from [37]). The fields were shaped by an add-on MLC for electrons presented in Figure 6.6. A minimum MLC field size of at least 3 cm × 3 cm is necessary for decomposing intensity distributions into leaf openings to ensure an output factor of nearly 1 and an energy-dependent penetration depth.

As a consequence, a treatment plan should consist of segment shapes satisfying certain constraints that ensure a minimum field size. For practical purposes it is also necessary that the field openings are connected and do not degenerate into two or more parts. These constraints have the consequence, that not every intensity matrix is decomposable into segments satisfying the constraints. This leads us to the task to find an approximation matrix and its decomposition into large segments, that differs from the given intensity matrix as few as possible. But indeed, this is not the only need of clinical practice. Actually, the physicians want good approximate segmentations of the target fluence, but with a limited number of different segments and a limited delivery time. As one can imagine, if the total change of a segmentation is minimal, there arise some segments that are not much larger than an $f \times w$-rectangle, which is unfavorable. To be precise, we somehow do not deal with the problem **Approx-MIN-TC**, but we have to minimize a linear combination of total change, number of segments and delivery time. But nobody knows how the three objectives have to be weighted. We just got the information that all segments should be "reasonably large" such that few of them are sufficient to build up competitive treatment plans.

As minimizing the number of segments is NP-hard, one cannot expect efficient algorithms for large problem instances and therefore heuristic approaches are required. Thus, we developed a heuristic segmentation algorithm that iteratively decides for a new segment and that takes into account that each of them is preferably large in order to reduce their quantity.
The final insight was, surprisingly, that the heuristic segmentation into large segments generates equivalent (or even slightly better) treatment plans as unconstrained segmentation, but enables a reasonable reduction in the segment number and monitor units, respectively (due to the large size of the single segments). The reason is that the unconstrained model with all its disadvantages described above does not produce bad treatment plans, but achieves its quality by incorporating many very small leaf openings increasing the number of segments and the delivery time.
We therefore now formulate a heuristic approach for **Approx-MIN-TC** with $\mathcal{S}' = \mathcal{S}_L$ with the extra constraint that each segment is reasonably large (without saying in detail, what the objective is exactly). This is maybe mathematically not satisfactory, but our experiences show that it gives no improvement for clinical practice if we put information into our model that were not given by the specialists.
The decomposition algorithm was implemented into an optimization programme in order to examine the assumptions of the algorithms for a clinical example. As a result, identical dose distributions (compared to exact segmentation) with much fewer segments and a significantly smaller number of monitor units could be achieved using dosimetric constraints. Consequently, the dose delivery is more efficient and less time consuming.
Let suitable parameters b_ℓ, b_r, w, h and f as described in Section 3 be given (for an illustration cf. Figure 3.2 in Section 3). Obviously, for the parameter set $b_\ell = n$, $b_r = 1$, $w = h = f = 1$, large segments are simply connected segments in the sense of Equation (2.1), the approximation problem has 0 as value of the objective function and degenerates to a segmentation problem into connected segments defined by Equation (2.1).

Remark 6. There already exists another heuristic segmentation algorithm for total change minimization with minimum field size constraint developed by Gauer and Kiesel (see [53]). There, a slightly more general definition of MFC-segments (called ASAS-segments there) is used. Every MFC-segment in the sense of this thesis is also an ASAS-segment there (but not vice-versa). As the new heuristics produces better results and is less complicated, we decided to use this one from now on. At the end of the section, we will provide numerical results comparing both heuristics.

6.4.1 Heuristic segmentation algorithm

The *h-w-environment* of a bixel $(i, j) \in [m] \times [n]$ is the area $[i - h + 1, i + h - 1] \times [j - w + 1, j + w - 1]$. Recall, that we artificially enlarged the segments to the area $[-h + 1, m + h] \times [-w + 1, n + w]$, when we defined the MFC-segments in Section 3. Thus, the bixels in the h-w-environment of (i, j) are the bixels that can be in a common $h \times w$-rectangle with (i, j). A bixel (i, j) is said to be *supplied*, if either $a_{ij} = 0$ or there

is an $h \times w$-rectangle comprising (i,j), such that $a_{k\ell} > 0$ for all the bixels (k,l) of the rectangle.
In order to reduce the complexity of the computation later on, we start with a heuristic preprocessing step, that decides for each bixel (i,j), if it is allowed to put $s_{ij} = 1$ in some segment S of the segmentation or not. To store this information, we use a matrix $Z = (z_{ij})_{(i,j)\in[m]\times[n]}$ that is defined as follows:

$$z_{ij} := \begin{cases} 0, & \text{if } a_{ij} = 0 \text{ and all bixels in the } h\text{-}w\text{-environment of } (i,j) \text{ are supplied,} \\ 1, & \text{otherwise.} \end{cases}$$

If $z_{ij} = 0$, this is a forced zero and we will claim $s_{ij} = 0$ for all segments in the segmentation. This is a reasonable choice, as all positive entries of the matrix in the h-w-environment of (i,j) can be covered by other rectangles where the matrix entries are positive. So it will never be a good idea to open (i,j) in order to irradiate a bixel in its h-w-environment.
Algorithm 8 describes the basic segmentation algorithm, that follows the preprocessing step.

Algorithm 8 Heuristic segmentation algorithm for segmentation into large segments

Input: intensity matrix A
  $done = 0$
  $Segmentation = \emptyset$
  while $done = 0$ **do**
    Find L-Segment: Compute an L-segment S with $\|A - S\|_1 < \|A\|_1$ such that $s_{ij} = 0$ whenever $z_{ij} = 0$ or return $done = 1$.
    if segment S found **then**
      $A = A - S$
      $Segmentation = Segmentation \cup S$
    end if
  end while
Output: *Segmentation*

The **Find L-Segment**-procedure (which is precisely described in Algorithm 12 in the appendix) has two (partly conflicting) objectives:

1. In each step, the value $\|A\|$ measures the total change corresponding to the current segmentation. Thus, the reduction $\|A\| - \|A - S\|$ should be maximized.

2. It is reasonable to choose leaf positions ℓ_i and r_i such that $a_{i,\ell_i} > a_{i,\ell_i-1}$ and $a_{i,r_i} > a_{i,r_i+1}$. As always, we set $a_{i0} = a_{i,n+1} = 0$ for all $i \in [m]$. For example, if A is $\begin{pmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{pmatrix}$, $w = 3$ and $h = 2$ the choice $S = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$ leads to the residual matrix $\begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$ that can be decomposed exactly. In

contrast, the choice $S = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$ is not optimal, as the residual matrix $\begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$ cannot be decomposed without positive total change.

As the ones of each segment can be covered by $h \times w$-rectangles, we can represent each segment by a union of rectangles of ones of height h. Thus, before we choose the next segment, we compute for each row $i \in [m-h+1]$ the set of "useful rectangles" of the form $[i, i+h-1] \times [\ell, r]$. Each such rectangle, determined by (i, ℓ, r), is evaluated by two numbers:

- the plus ratio $p^+ := \frac{|\{(k,j):\ a_{kj}>0 \ \mid\ k\in[i,i+h-1],\ j\in[\ell,r]\}|}{(r-\ell+1)h}$, giving the ratio of positive matrix entries in the rectangle and
- the up-down ratio $u := \frac{|\{k\in[i,i+h-1] \ \mid\ a_{k,l}>a_{k,\ell-1}\}|+|\{k\in[i,i+h-1] \ \mid\ a_{k,r}>a_{k,r+1}\}|}{2h}$, measuring the ratio of good leaf positions in the sense of the second objective described above.

We call a rectangle (i, ℓ, r) a *useful* rectangle if

- $r - \ell + 1 \geq w$, $\ell \leq b_\ell$, $r \geq b_r$ according to the requirements of an L-segment,
- its plus ratio p^+ is larger or equal than a given parameter p,
- if $r - \ell + 1 > w$ then neither $(a_{i,\ell}, \ldots, a_{i+h-1,\ell})$ nor $(a_{i,r}, \ldots, a_{i+h-1,r})$ is the zero vector and
- no forced zero is comprised in $[i, i+h-1] \times [\ell, r]$.

The set of useful rectangles in row i is denoted by $\mathcal{R}_i$ and consists of tuples of the form (ℓ, r, p^+, u). In the Find L-Segment-procedure, we will build a segment S iteratively by choosing useful rectangles that we add to S. Adding a rectangle R to S means setting $s_{ij} = 1$ for all $(i, j) \in R$. A rectangle R is called *S-feasible*, if S is still connected and satisfies conditions (i)-(iii) of the definition of an MFC-segment (cf. Section 3) after adding R to S.
The segmentation step is then computed as described in Algorithm 12 in the appendix. In line 2 and 3 of the algorithm, we compute the first respectively last row where it makes sense to open a rectangle of height h. Of course, there must exist a useful rectangle in these rows and it is not a good idea to irradiate any rows of A that are already smaller or equal to zero in each column. The while-loop in line 8 makes sure that we search for a segment until we find one improving the total change by trying different start rows. Having decided on a start row i_{start}, we choose the best rectangle in this row with regard to our objective function, that is a linear combination of plus ratio and up-down ratio. The weight of these two numbers in the objective function depends on the given parameter λ. Then we go through the matrix row by row and in each row we decide for opening another rectangle or not. If $i \leq i_{end}$ and there is an S-feasible rectangle, we choose one with best objective value. If not and $\boldsymbol{s}_i \neq \mathbf{0}$ or there are already f opened rows, we may also open no rectangle. If $\boldsymbol{s}_i = \mathbf{0}$ and there

are not f opened rows, we choose the leaf positions of row $i-1$ also for row i and go on iterating. If we arrive at a step where $r_{i-1} - l_{i-1} < w - 1$ (that means we decided to finish the segment when we checked row $i-1$) or ($i > i_{end}$ and enough rows are open), we finish the segment. If the chosen segment improves the total change, we are satisfied. If not, we try a new reasonable start row.
Note that deciding for an S-feasible rectangle in a row also ensures that the zeros of the resulting segment can be covered by $h \times w$-rectangles (cf. the definition of S-feasible). If $\ell_i < \ell_{i-1}$ or $r_i > r_{i-1}$, we check in the rows $i-h, \ldots, i-1$ if the resulting zero regions are really large enough.
The procedure described above depends on two parameters, namely the plus ratio bound p defining useful rectangles and the weight λ defining the importance of the plus-ratio compared to the up-down ratio in the objective for each rectangle. Computational tests have shown that $p \in \{0.5, 0.6, 0.7\}$ and integer values $\lambda \in [5, 13]$ lead to good results. That is why we compute the segmentation algorithm for each combination of p and λ and choose the best segmentation we get.

Remark 7. To explain that the problem we have to solve here for clinical practice is indeed not **Approx-MIN-TC** we did some more computational tests. We defined $\mathcal{S}'$ to be the set of all rectangular segments of size at least $f \times w$ satisfying the leaf overtravel constraint and solved **Approx-MIN-TC** for this set of segments using an integer linear programming approach similar to the one described in Section 6.1.1 (without the $\leq C$-constraints). We solved this integer program using Gurobi [59] as ILP solver. Each rectangle is an L-segment and we get a feasible solution of **Approx-MIN-TC** with $\mathcal{S}' = \mathcal{S}_L$. The total change values for our test instances were significantly smaller than the ones produced by our heuristic, but at the expense of a huge number of segments. These results are totally useless for treatment plans and we indeed need an algorithm taking the different objectives into account. To illustrate this effect, we added these results to our analyses in Section 6.4.3.

6.4.2 Clinical case

A clinical case was set-up to examine the efficiency of our proposed segmentation algorithm. For a patient with cancer of the right breast, electron irradiation plans were created with a self-designed IMRT optimization programme based on our previous studies [25, 27]. The planning target volume was the right breast, which should receive a total dose of 50.4 Gy (1.8 Gy per fraction). In addition, the target volume should be covered by the 95% isodose line (95% of the prescribed dose). The ipsilateral lung was considered to be organ at risk.
The optimization programme provides simultaneous optimization of beam orientation, energy and intensity for dose delivery with an add-on MLC for electrons (Euromechanics, Schwarzenbruck, Germany) presented in Figure 6.6 and [35, 37]. Electron dose calculation was performed by Monte Carlo simulations with the treatment planning system *Pinnacle* from Philips (Version 8.1s). Final dose calculation of the treatment plans was conducted using a dose grid size of 3 mm and a dose calculation uncertainty of 2%.

The segments from the segmentation are treated as candidates for the treatment plan. In a final optimization step, the dose of the candidates is calculated for all beam energies and then optimized for a given weight proportion between best target coverage and minimum dose to critical organs in order to find the final set of segments with optimal beam energies and their corresponding monitor units.
This final step justifies our approximative approach in the segmentation, as a larger approximation error does not necessarily result in a suboptimal treatment plan. Indeed, larger segments produce homogeneous dose distributions and thus, the same final fluence can be generated using fewer larger segments. The acceptability of a treatment plan is decided by means of dose volume histograms (cf. Section 6.4.3) and a plan is only presumed if the required dose constraints are not exceeded. Therefore, the danger of cumulative deviation in the approximation step does not really exist, as the computed segments are just candidates for the treatment plan that pass through a further optimization step.

6.4.3 Results

First, we compare electron IMRT plans created with different segmentation settings for the clinical case prescribed in Section 6.4.2. Finally, we give a detailed evaluation for the results of the decomposition step.
A treatment plan with a segmentation setting fwh uses the heuristic decomposition algorithm with a minimum total field height of f, a minimum rectangle width w and a minimum rectangle height h. The decomposed matrices vary in their vertical size m and their horizontal size n, as they describe only parts of the beam head where the target volume is located. Thus, in practice, the overtravel parameters b_l and b_r will depend on the positioning of the matrix and are put individually for each matrix. Our electron MLC is capable of shifting the leaf edges to 3/4 of the radiation field.

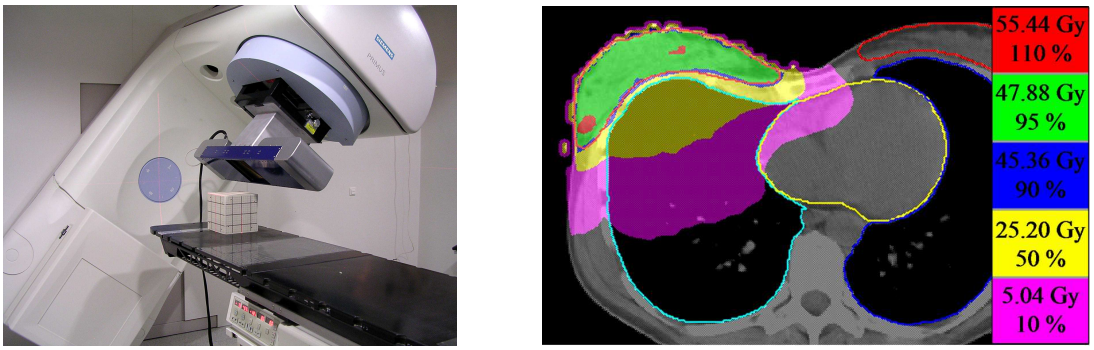

Fig. 6.6: MLC and example of a dose distribution. Left: Add-on MLC for electrons mounted on a conventional Siemens accelerator. Right: Dose distribution of an electron IMRT plan consisting of 51 MLC fields achieved through segmentation setting 442. The corresponding dose volume histogram is shown in Figure 6.7. The setting 442 is given by a minimum total field height $f = 4$, a minimum rectangle width $w = 4$ and a minimum rectangle height $h = 2$.

The plan quality was evaluated by means of dose volume histograms that indicate the amount of dose delivered to a certain volume of the patient (here: the right breast and

the right lung). Thus, dose homogeneity in the target volume and dose exposure to the organs at risk can be examined. In Figure 6.7, the dose volume histogram demonstrates that identical target coverage can be achieved using smaller or larger minimum MLC openings, while the sparing of the organs at risk can be improved using larger fields (cf. setting 111 and 441). In fact, the treatment plan performed out to be less liable to breathing motion of the patient by the use of the parameter $h = 2$ which avoids single leaf openings and closings. A physical explanation for this effect can be found in [36]. The quality of the treatment plan is almost equivalent for the settings 441 and 442. Table 6.1 illustrates the main benefit of our approach with the dosimetric constraints. The setting 441 enables a better treatment quality than setting 111 and this can be achieved with much fewer segments (36 instead of 83) and a significantly smaller number of monitor units (15516 instead of 54167). As a result, the dose delivery is more efficient and less time consuming. The setting 442 still provides an acceptable number of segments and monitor units and thus should be favored due to the robustness of the plan mentioned above. As the leaf width is 0.7 cm, fields with a horizontal and vertical height of 4 bixels have a size of approximately 3 cm × 3 cm and this confirms our dosimetric constraint of 3 cm × 3 cm minimum segment size (cf. Figure 6.5). It can also be demonstrated that minimum segment sizes greater than setting 442 do not improve the plan, because the dose volume histograms were considerably better when using minimum segment sizes smaller than setting 552.

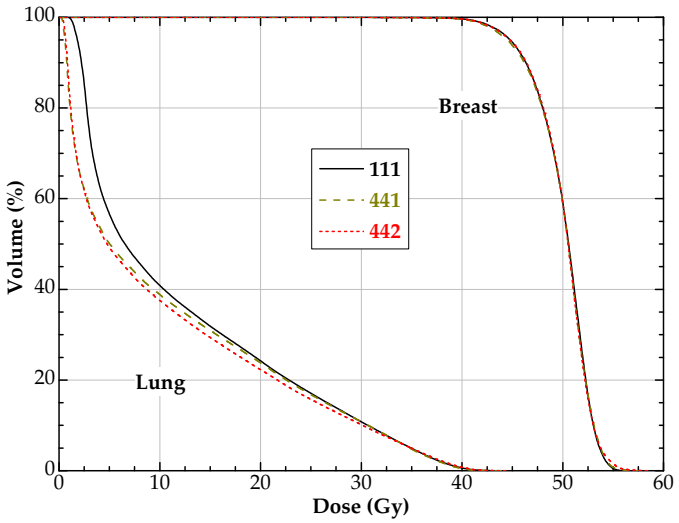

Fig. 6.7: The dose volume histogram for settings 111, 441 and 442 demonstrates that identical (or even better) results could be achieved when using greater minimum segment sizes (cf. 111 and 441) and segment shapes with larger vertical open and closed areas (cf. 441 and 442). The setting fwh is given by a minimum total field height f, a minimum rectangle width w and a minimum rectangle height h. The resulting dose distribution for setting 442 is presented in Figure 6.6.

It is important to emphasize that the number of segments and the number of monitor units in Table 6.1 belong to the final IMRT plan and result from the third optimization

step and not from the decomposition step of our algorithm. In fact, the monitor units have another scale here and are not directly comparable with the delivery time from the segmentation. In contrast, the total change information stems from the decomposition step. Note, that the total change of the segmentation itself is not a significant quantity, because if the matrix entries are large, a larger total change is acceptable. Therefore, we compute the total sum of entries for each intensity matrix and then calculate the relative total change which is the ratio between total change and total sum of entries. The smaller the relative total change, the better is the decomposition.

Tab. 6.1: Segmentation results of IMRT plans using different decomposition settings. Setting xyz means $f = x$, $w = y$ and $h = z$.

Setting	Mean Relative Total Change	Number of Segments	Number of Monitor Units
111	0.04	83	54167
221	0.16	70	42556
222	0.21	71	52783
331	0.26	65	35101
332	0.27	64	31473
333	0.31	52	24616
441	0.31	36	15516
442	0.30	51	21865
443	0.35	36	12719
444	0.38	36	11672
551	0.37	29	11904
552	0.34	38	14466

For the detailed evaluation of our algorithms, we use our set of 475 clinical intensity matrices that originate from electron treatment plans for different patients and beam angles. The matrices are produced in the pre-segmentation step of the treatment planning (see [25, 27]). Exemplarily, we compute segmentations for the parameters $f = w = h = 3$. The leaf overtravel constraint is neglected here. We compare the values of total change, delivery time and number of segments produced by four different segmentation algorithms:

1. **Our heuristics:** The heuristic segmentation algorithm described in Section 6.4.1.

2. **Old heuristics:** The older heuristic segmentation algorithm with a slightly more general model of segments described in [53].

3. **Column generation:** The column generation approach for **Approx-MIN-TC** with subproblem for L-segments with $h > 1$ described in Section 6.3.

4. **Rectangles:** The integer linear programming approach for **Approx-MIN-TC** with $\mathcal{S}'$ being the set of rectangular segments of size at least $f \times w$ (described in Remark 7).

The results (averaged over the 475 matrices) are shown in Table 6.2. As a treatment plan is a superposition of several intensity profiles from different beam angles, the

approximation errors balance each other and lead to applicable treatment plans as described above. The results for the old and the new heuristics could be produced in few minutes on a 2.5GHz workstation. The column generation results took a computation time of approximately 11 hours. The rectangle approach took about 7 hours.

Tab. 6.2: Numerical results for L-segments with $f = w = h = 3$.

	Our heuristics	Old heuristics	Column generation	Rectangles
Total change	443.15	483.89	412.62	289.59
Relative total change	0.30	0.32	0.30	0.21
Delivery time	31.13	26.30	18.44	115.41
Number of segments	27.02	21.20	10.02	69.33

Finally, it is interesting which one of the first three algorithms performed best in how many cases. For our 475 matrices the column generation produced the smallest total change in 313 cases. The new heuristics was best 124 times and the old heuristics in 38 cases. Basically, the new heuristics outperforms the old one and leads to much smaller total change values. But there is a (still acceptable) increase of the delivery time and the number of segments compared to the old heuristics.
In conclusion, the heuristics we described here is capable of producing good segmentation results that can be used for efficient treatment planning with high quality. The histograms show that the use of larger segments results in IMRT plans with fewer segments and monitor units respectively. Although the approximation error of the segmentations rises with increasing minimum field size, equivalent or even better dose distributions could be achieved. Furthermore, the larger segment sizes lead to plans that are less liable to breathing motion of the patient. Concluding, this approach to approximated segmentation in IMRT planning shows the potential of these ideas and there is a need for further research in related approximation problems. The column generation approach produces even better segmentation results, but is not applicable in clinical practice because of very long computation times and thus it is just of mathematical interest. The rectangle approach shows how small the total change could be if the problem to solve was **Approx-MIN-TC**, but it is useless for the needs of treatment planning.

7. APPROXIMATE CONTINUOUS SEGMENTATION

The discrete segmentation model has a number of disadvantages that lead to an inexact model of fluence distributions. This section embeds the segmentation task into the wider context of function approximation and models both profiles and segments as real-valued functions of two variables. This leads to convex optimization problems whose objective is to minimize the approximation error between the profile and the superposition of the real-weighted segments. Thus, a more realistic model of radiation is used and may enable an improvement in treatment quality.
The discrete segmentation model makes a number of basic model assumptions like:

- Radiation behaves linearly. The fluence distribution of a superposition of two segments with disjoint ones is equivalent to the intensity of their sum (additivity). The fluence distribution of λS is λ times the fluence distribution of S for each shape S (homogeneity).

- The intensity is 0 in covered regions and 1 in uncovered regions of the field. Especially, we have a 0-1-step at the boundary of the rectilinear polygon. As a consequence, the output factors do not depend on the field size, i.e. the maximal dose output delivered by a field is always 1 for all field shapes.

Whereas the first assumption is (at least) approximately true in physical reality, the second one does not hold for a variety of physical reasons (see Figure 6.5). In fact, there is a penumbra at the boundary of the shapes that is not modeled in these formulations. Thus, we observe a decline of intensity from 1 to 0 having a certain width and decline curve. In addition, small fields have a lower dose output and the intensity in the middle of the field is smaller than 1. We keep the linearity assumption in our following approach, but also give an alternative approach in Section 7.5 using column generation that is capable of dealing with a non-additive intensity model and also takes intraleaf-leakage into account.
The lack of the discrete model is the missing realization of the penumbra of the radiation. Furthermore, the model is also rather restrictive, because in clinical practice the radiation field is not discrete in horizontal direction and the leaves can be positioned with an accuracy of $0.01\,cm$. Besides, the monitor units have to be integers in the discrete model. In practice, the irradiation time can be set up in steps of 0.1 monitor units (data provided by the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf).
Therefore, we formulate the segmentation problem in a wider context of function approximation, where we associate with each MLC leaf positioning a segment that is no longer a 0-1-matrix, but a real-valued function of two variables. The segments

have the property that they are 1 in some specified region, have a certain monotone decline to 0 at the boundary of this region and have the value 0 in the rest of the field. Using the linearity assumption, we show that complex fields can be represented by a sum of small fields. Moreover, small fields have lower output factors than 1 in our model and only their superposition to larger shapes yields a dose output of 1 which fits to the observation from experiments. Thus, the new approach provides a more realistic model of the reality. The desired intensity profiles are given by real-valued functions f of two variables. The task is then to find a decomposition $\sum_{S\in\mathcal{S}'} u_S S$ with nonnegative coefficients u_S that minimizes $\|f - \sum_{S\in\mathcal{S}'} u_S S\|$, where $\mathcal{S}'$ is the set of deliverable segments, that we want to use for the optimization.

In clinical application, our approach is justified, as the whole treatment planning process underlies uncertainties like inexact dose calculation and suboptimal choice of beam-angles. Furthermore, during the treatment itself there might occur underdosage and overdosage effects due to the patient's motion. Therefore, it makes sense to improve the accuracy in delivering the target fluence with an improved segment model. It is important to mention that the segment model is not only used in the segmentation step, but also to improve the dose calculation for the chosen segments in the optimization steps of the whole treatment planning process after the segmentation (cf. [25] for details).

One can show that **Approx-MIN-TC** can be regarded as a subcase of this continuous problem. However, the new approach does not substitute the approaches for the discrete problems, as our method still makes use of discrete decomposition algorithms.

In this section, we solve the above mentioned continuous segmentation problem under some simplifying assumptions. It is important to mention that the task is far from being completely solved and many problems result from our considerations. Thus, we will mention some generalizations and extensions that will be topics for further research. Besides, the approach needs more investigations in order to fit the degrees of freedom to the reality, before we can expect a benefit for the clinical practice.

7.1 Definitions and problem formulation

Throughout this Section 7 of the thesis, different from the previously introduced notation, intervals $[v, w]$ with $v, w \in \mathbb{R}$ denote continuous real-valued intervals. For integral intervals $[k, \ell]$ with $k, \ell \in \mathbb{Z}$ we will use the notation $\{k, \ldots, \ell\}$.

We assume the number of leaf pairs of the MLC to be m and the irradiated area to be the rectangle $[0, m] \times [0, n]$. Let the desired intensity profile be given as a function $f : [0, m] \times [0, n] \to \mathbb{R}_+$ of two variables. Maybe we have given f by a number of function values $f(x_i, y_i)$ of supporting points (x_i, y_i) with $(x_i, y_i) \in [0, m] \times [0, n]$. In this case, we can use (e.g. bilinear) interpolation to extend f.

The segments are also modeled as real-valued functions $S : [0, m] \times [0, n] \to [0, 1]$ and $S(x, y)$ determines the fluence that is delivered to the position (x, y). Having specified a set of feasible segment functions $\mathcal{S}'$, we have to choose a norm $\|\cdot\| : f \mapsto \|f\|$ to

measure the quality of the approximation. The resulting optimization problem is:

$$\textbf{Approx-MIN-TC-Continuous:} \qquad \Big\|f - \sum_{S\in\mathcal{S}'} u_S S\Big\| \to \min \tag{7.1}$$

subject to $u_S \geq 0$ for all $S \in \mathcal{S}'$. If one additionally requires a minimum irradiation time for each segment, one has to add the condition $u_S \geq \delta$ for all $S \in \mathcal{S}'$ and some threshold $\delta \geq 0$. In analogy to the discrete problems, the value of the objective function is called *total change* (TC). In analogy to the term "discrete segmentation" we will speak of continuous segmentation methods in this context.
In the next sections we explain how the set $\mathcal{S}'$ is chosen and solve the problem **Approx-MIN-TC-Continuous** using the L_2-norm as measure for the approximation quality. This leads us to a constrained quadratic programming (QP) problem. Before we describe the modeling of the segment functions, we impose a number of characteristics the segments have to satisfy: at first some linearity constraints that we need to make our QP-approach work and then some constraints that we need to model the physical behavior of radiation.
Therefore, we define a map between MLC leaf positions $((\ell_1, r_1), \ldots, (\ell_m, r_m))$ and the corresponding fluence distributions S. We use the expression shape for the leaf positions, while the corresponding fluence distributions are called segments in this section. We distinguish between the uncovered area $O = \bigcup_{i\in\{1,\ldots,m\}}[i-1, i] \times [\ell_i, r_i]$ of a shape, that is not covered by the leaves and the covered area $C = [0, m] \times [0, n] \setminus O$. By definition $O \cup C = [0, m] \times [0, n]$. The segment function that corresponds to the uncovered area $O \subseteq [0, m] \times [0, n]$ is denoted by S_O.

7.1.1 A linear model of segments

Throughout, we assume that irradiating a shape with fluence distribution S for a time $t \in \mathbb{R}_+$ yields a fluence distribution of tS. Thus, homogeneity is assumed and we only have to model the fluence distribution that corresponds to irradiating a certain shape for a unit time of 1.

Definition 8 (Additive model of fluence distributions)**.** A *model of fluence distributions* is a map that assigns to each uncovered area $O \subseteq [0, m]\times[0, n]$ a segment function $S_O : [0, m] \times [0, n] \to [0, 1]$. A model of fluence distributions is called *additive* if for each pair of uncovered areas O and O' that do not have a common interior point, we have

$$S_O(x, y) + S_{O'}(x, y) = S_{O\cup O'}(x, y) \text{ for all } (x, y) \in [0, m] \times [0, n]. \tag{7.2}$$

We now make two basic model assumptions that characterize the fluence distributions and show that these ensure the additivity of the model:

1. There exist a family of functions $\{S'_{[u,v]} : [0, m] \to [0, 1], \quad 0 \leq u \leq v \leq m\}$ and a family of functions $\{S''_{[\ell,r]} : [0, n] \to [0, 1], \quad 0 \leq \ell \leq r \leq n\}$ such that the fluence distribution for all rectangular uncovered areas $[u, v] \times [\ell, r]$ with $0 \leq u \leq v \leq m$ and $0 \leq \ell \leq r \leq n$ is given by

$$S_{[u,v]\times[\ell,r]}(x, y) = S'_{[u,v]}(x) \cdot S''_{[\ell,r]}(y) \text{ for all } (x, y) \in [0, m] \times [0, n]. \tag{7.3}$$

2. The functions S' and S'' are additive with respect to their open intervals, i.e. if $[\ell, r], [r, s] \subseteq [0, n]$ and $[u, v], [v, w] \subseteq [0, m]$, we require

$$S'_{[u,v]}(x) + S'_{[v,w]}(x) = S'_{[u,w]}(x) \text{ for all } x \in [0, m], \tag{7.4}$$
$$S''_{[\ell,r]}(y) + S''_{[r,s]}(y) = S''_{[\ell,s]}(y) \text{ for all } y \in [0, n]. \tag{7.5}$$

As the uncovered region of the MLC is a rectilinear polygon with a discretization of step width 1 in vertical direction, for each uncovered region O we can find a partition $O = R_1 \cup \cdots \cup R_t$ of O into rectangles. Using the fluence distributions S_{R_i}, $i \in \{1, \ldots, t\}$, for the rectangular uncovered regions, we define the fluence distributions S_O for an arbitrary shape with uncovered region $O = \bigcup_{i=1}^{t} R_i$ by

$$S = \sum_{i=1}^{t} S_{R_i}. \tag{7.6}$$

Theorem 27. *If the segment functions satisfy (7.3), (7.4) and (7.5), then the segment functions are well-defined by (7.6) and the model of fluence distributions is additive.*

Proof. Obviously, it is sufficient to show, that for $u, v, w \in [0, m],\ u \leq v \leq w$ and $\ell, r \in [0, n],\ \ell \leq r$ the equation

$$S_{[u,v]\times[\ell,r]}(x, y) + S_{[v,w]\times[\ell,r]}(x, y) = S_{[u,w]\times[\ell,r]}(x, y) \tag{7.7}$$

holds for all $(x, y) \in [0, m] \times [0, n]$. Thus, the rectangular fluence distributions are additive, the definition for the rectangles does not contradict Equation (7.6) and the fluence distribution of an arbitrary shape does not depend on the partition into rectangles. All in all, everything is well-defined and additivity is given. Now let us show Equation (7.7). Let $S'_{[u,v]}$, $S'_{[v,w]}$ and $S''_{[\ell,r]}$ be such that

$$S'_{[u,v]}(x) \cdot S''_{[\ell,r]}(y) = S_{[u,v]\times[\ell,r]}(x, y)$$

and

$$S'_{[v,w]}(x) \cdot S''_{[\ell,r]}(y) = S_{[v,w]\times[\ell,r]}(x, y)$$

for all $(x, y) \in [0, m] \times [0, n]$. Then, we have

$$\begin{aligned} S_{[u,v]\times[\ell,r]}(x, y) + S_{[v,w]\times[\ell,r]}(x, y) &= S'_{[u,v]}(x) \cdot S''_{[\ell,r]}(y) + S'_{[v,w]}(x) \cdot S''_{[\ell,r]}(y) \\ &= (S'_{[u,v]}(x) + S'_{[v,w]}(x)) \cdot S''_{[\ell,r]}(y) \\ &= S'_{[u,w]}(x) \cdot S''_{[\ell,r]}(y) = S_{[u,w]\times[\ell,r]}(x, y) \end{aligned}$$

and the proof is complete. □

7.1.2 Modeling of the physical behavior of radiation

Firstly, we impose the constraint that horizontal and vertical fluence are equal, i.e.

$$S''_{[\ell,r]}(y) = S'_{[\ell,r]}(y) \tag{7.8}$$

for all appropriate values of ℓ, r and y.
Now we model the decline of intensity at the border of the uncovered region of the MLC. Therefore, we make more model assumptions on the segments and use monotone functions that increase from zero to one for modeling the penumbra.

Definition 9 (Decline function)**.** A real-valued function $t : \mathbb{R} \to [0,1]$ is called a *decline function* if $t(x) = 0$ for $x \leq -\frac{1}{2}$, $t(x) = 1$ for $x \geq \frac{1}{2}$, t is increasing in $[-\frac{1}{2}, \frac{1}{2}]$ and point symmetric with respect to the point $(0, \frac{1}{2})$.

In the 0-1-model of segmentation, one assumes that the intensity decline at the boundary of a segment is according to the one-step function

$$t_1(x) = \begin{cases} 0, & x < 0, \\ \frac{1}{2}, & x = 0, \\ 1, & x > 0. \end{cases} \tag{7.9}$$

A function that models the reality better seems to be a piecewise linear function with 3 sectors, i.e.

$$t_2(x) = \begin{cases} 0, & x < -1/2, \\ x + 1/2, & -1/2 \leq x < 1/2, \\ 1, & x \geq 1/2. \end{cases} \tag{7.10}$$

A piecewise quadratic decline function like

$$t_3(x) = \begin{cases} 0, & x < -1/2, \\ 2(x+1/2)^2, & -1/2 \leq x < 0, \\ -2(x-1/2)^2 + 1 & 0 \leq x < 1/2, \\ 1, & x \geq 1/2 \end{cases} \tag{7.11}$$

seems to be a good choice as a steep descent in the middle of the decline region fits to the observation from reality.
Each of the decline regions can be expanded to the interval $[-\frac{\gamma}{2}, \frac{\gamma}{2}]$ for an arbitrary decline width $\gamma \geq 0$ by setting

$$t^\gamma(x) = \begin{cases} 0, & x < -\gamma/2, \\ t\left(\frac{x}{\gamma}\right), & -\gamma/2 \leq x < \gamma/2, \\ 1, & x \geq \gamma/2. \end{cases} \tag{7.12}$$

The linear decline function converges to the one-step function as $\gamma \to 0$. The functions (7.9), (7.10) and (7.11) are illustrated in Figure 7.1.

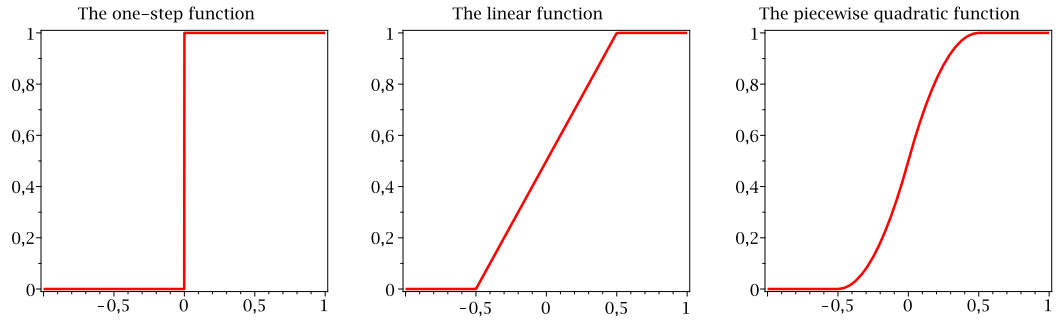

Fig. 7.1: Three different decline functions.

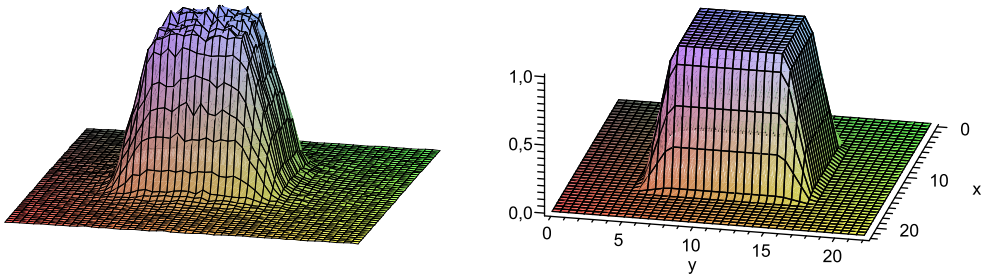

Fig. 7.2: The segment on the left is a realistic fluence distribution of a field with size 10 $cm \times 10$ cm, while on the right a rectangular segment with linear decline function is modeled. The comparison shows that our model fits quite well to real fluence distributions.

Dose calculations using Monte-Carlo simulations are the standard that the medical physics community considers to give the actual behavior of radiation. Figure 7.2 on the left shows the fluence distribution of a 10 $cm \times 10$ cm field. The data was provided by the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf and shows that an approximation with linear or quadratic decline would be appropriate and sufficient. Figure 7.2 on the right shows a rectangularly modeled segment with $m = n = 22$, the uncovered region $M = [6, 16] \times [6, 16]$ and a linear decline function with width $\gamma = 2$.
Let the decline function $t : \mathbb{R} \to [0, 1]$ and the decline width $\gamma > 0$ be fixed. Using our model of decline functions, we assume that the left leaf at position ℓ and the right leaf at position r provide a decline region of width γ and the segment function is:

$$S'_{[\ell,r]}(x) = t\Big(\frac{x-\ell}{\gamma}\Big) - t\Big(\frac{x-r}{\gamma}\Big). \tag{7.13}$$

Note, that if $r - \ell \geq \gamma$, this reduces to

$$S'_{[\ell,r]}(x) \quad := \quad \begin{cases} t\big(\frac{x-\ell}{\gamma}\big), & \text{if } x < \ell + \frac{\gamma}{2} \\ 1, & \text{if } \ell + \frac{\gamma}{2} \leq x \leq r - \frac{\gamma}{2} \\ t\big(\frac{r-x}{\gamma}\big), & \text{if } x > r - \frac{\gamma}{2}, \end{cases} \tag{7.14}$$

i.e. we have an increase in intensity from 0 to 1 in $[\ell - \gamma/2, \ell + \gamma/2]$ and a decrease from 1 to 0 in $[r - \gamma/2, r + \gamma/2]$. The middle points of the decline regions coincide with the positions of the MLC leaves.
Figure 7.3 on the left shows the functions $S'_{[\ell,r]}$ for $\ell = 2$ and four different values of r with quadratic decline function and $\gamma = 2$. Obviously, if the decline regions overlap, the full-dose region vanishes and the maximum output factor is smaller than 1. On the right, an example for a one-dimensional segmentation with segments of type $S_{[i-1,i]}$, quadratic decline and $\gamma = 2$ is shown.

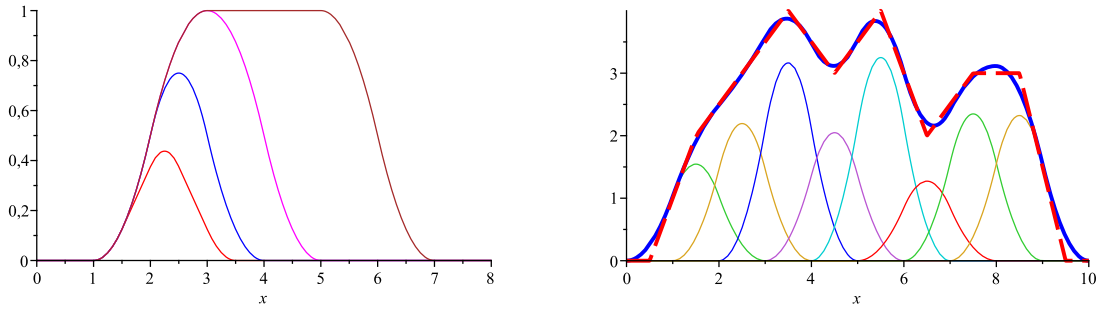

Fig. 7.3: Example of a one-dimensional segmentation. The left figure shows the segments $S'_{[\ell,r]}$ for $\ell = 2$ and $r \in \{2.5, 3, 4, 6\}$ with quadratic decline function and $\gamma = 2$. An exemplary segmentation with segments of type $S_{[i-1,i]}$ is shown on the right. The dashed curve is the target fluence distribution f, that results from linear interpolation of the points $(0,0), (0.5,0), (1.5,2), (2.5,3), (3.5,4), (4.5,3), (5.5,4), (6.5,2), (7.5,3), (8.5,3), (9.5,0)$ and $(10,0)$. The solid thick curve is the approximation and the other curves are the components $u_{S_{[i-1,i]}} S_{[i-1,i]}$ whose sum is the approximation.

It is an interesting observation, that our segment functions are basically the well-known cardinal B-Spline functions defined for each $m \in \mathbb{N}$ as:

$$N_1(x) := \begin{cases} 1, & 0 \le x < 1, \\ 0, & \text{otherwise}, \end{cases}$$

$$N_m(x) := (N_{m-1} \star N_1)(x) = \int_{-\infty}^{\infty} N_{m-1}(x-t) N_1(t) dt = \int_0^1 N_{m-1}(x-t) dt,$$

where $\star$ is the convolution operator. We then have the relation:

$$\begin{aligned} N_1(x) &= t_1(x) - t_1(x-1), \quad \text{(except for } x = 0 \text{ and } x = 1), \\ N_2(x) &= t_2(x - 1/2) - t_2(x - 3/2), \\ N_3(x) &= t_3((x-1)/2) - t_3((x-2)/2). \end{aligned}$$

That means, N_1 is the segment function for the discrete model and $[\ell, r] = [0, 1]$, N_2 is the segment function for linear decline with $\gamma = 1$ and $[\ell, r] = [1/2, 3/2]$ and N_3 is the segment function for quadratic decline with $\gamma = 2$ and $[\ell, r] = [1, 2]$.

The additivity of the one-dimensional model of fluence distributions above is immediately clear, because

$$\begin{aligned} S'_{[\ell,r]}(x) + S'_{[r,s]}(x) &= \Big(t\Big(\frac{x-\ell}{\gamma}\Big) - t\Big(\frac{x-r}{\gamma}\Big)\Big) + \Big(t\Big(\frac{x-r}{\gamma}\Big) - t\Big(\frac{x-s}{\gamma}\Big)\Big) \\ &= t\Big(\frac{x-\ell}{\gamma}\Big) - t\Big(\frac{x-s}{\gamma}\Big) \\ &= S'_{[\ell,s]}(x). \end{aligned}$$

The vertical and horizontal fluence distributions superimpose each other and the two-dimensional result can be factorized into a horizontal and a vertical component. The effect is shown in Figure 7.4.

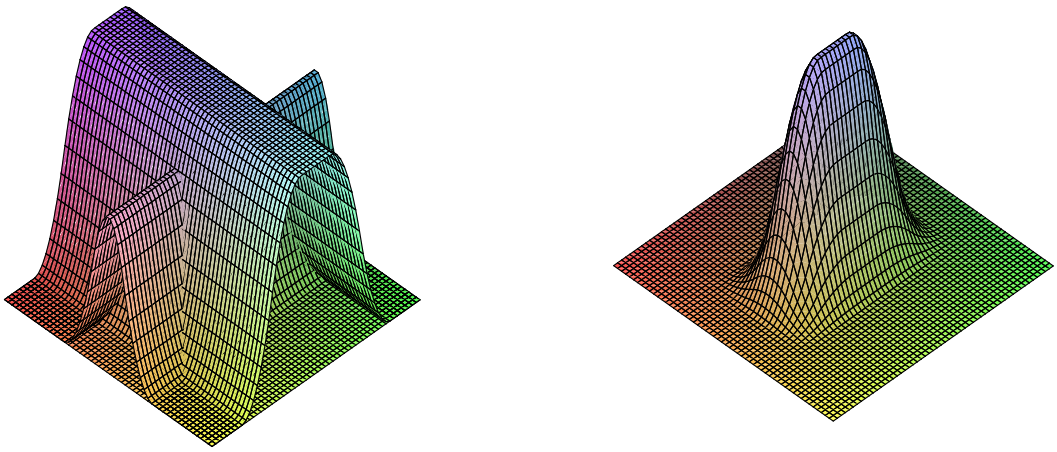

Fig. 7.4: Horizontal and vertical component of the segments. The fluence distribution corresponding to the penumbra left and right from the uncovered region as well as the fluence distribution for the penumbra above and below the uncovered region are shown in the left figure. Their product is the two-dimensional fluence distribution on the right figure.

As the borders of the rectilinear uncovered region coincide with the MLC leaves, our uncovered regions O are discrete in the vertical direction. In our quadratic programming approach that is discussed in Section 7.2, we discretize the irradiation field also in horizontal direction, that means we allow only leaf positions $((\ell_i, r_i))_{i\in\{1,\dots,m\}}$, where $\ell_i, r_i \in \{0,\dots,n\}$ for all $i \in \{1,\dots,m\}$. This means $\mathcal{S}'$ is the set of all segments corresponding to integral leaf positions. We do this to reduce the number of possible shapes as the complexity of our optimization problem is too large otherwise. However, some generalizations are worth to be considered. Using additivity, we can make $\mathcal{S}'$ even much smaller, as it is obviously sufficient to work with

$$\mathcal{S}' = \{S_{[i-1,i]\times[j-1,j]} \mid (i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}\}. \tag{7.15}$$

The area $[i-1,i]\times[j-1,j]$ is called bixel (i,j) in the following. Thus, $\mathcal{S}'$ is the set of segments that correspond to single bixel openings. We call them *basic segments*. Each complex shape can then be written as a sum of basic segments. Let us exemplarily have

a look at such fluence distributions with $\gamma = 2$ and linear and quadratic decline. The segments are shown in Figure 7.5 and one can see that the maximum fluence values are smaller than 1 because the decline regions overlap. The linear basic segment has a maximum of $\frac{1}{4}$ and the quadratic basic segment has a maximum of $\frac{9}{16}$.

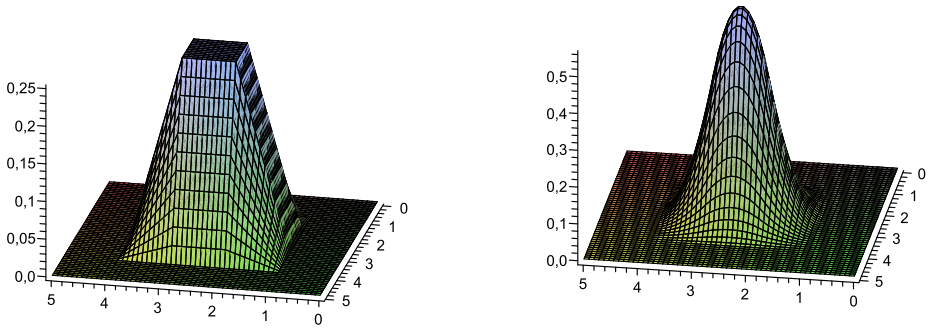

Fig. 7.5: Basic segments for linear and quadratic decline.

Finally, we show a more complex fluence distribution that is obtained by combining several rectangular regions, namely the rectangular regions of each open row of the MLC. An example is shown in Figure 7.6.

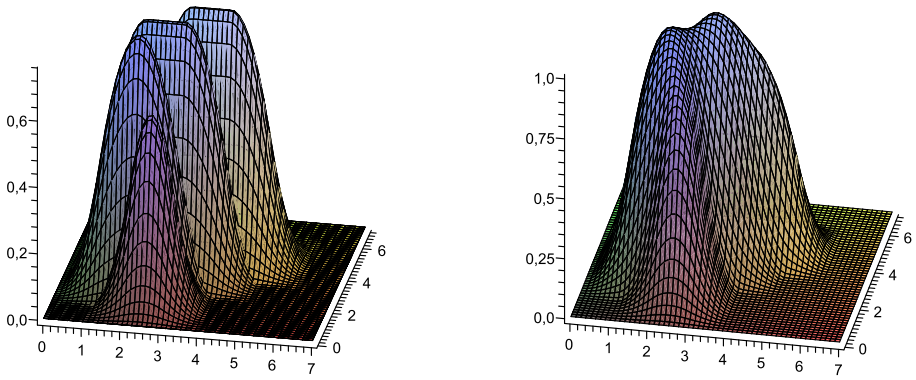

Fig. 7.6: Superposition of rectangular segments. For the leaf positions $((2,3),(1,3),(1,4),(2,5))$ we see the four fluence distributions (with quadratic decline) on the left and the fluence distribution of the combined shape, which is their sum, on the right.

7.2 Solution of the continuous segmentation problem

Now we solve **Approx-MIN-TC-Continuous**. Suppose, that the intensity profile is given by $f : [0,m] \times [0,n] \to \mathbb{R}_+$. Let $\mathcal{S}' = \{S_1, \ldots, S_t\}$ be the set of segments we want to consider, $S_k : [0,m] \times [0,n] \to [0,1]$ for all $k \in \{1, \ldots, t\}$. We use the L_2-norm to

measure the quality of the approximation, as this leads to a quadratic programming problem that can be solved fast and reliably. Our task is to minimize

$$g(\boldsymbol{u}){=}g(u_1,\ldots,u_t) = \Big\|f - \sum_{k=1}^{t} u_k S_k\Big\|_2^2 \tag{7.16}$$

$$= \int_0^m \int_0^n \big(f(x,y) - \sum_{k=1}^{t} u_k S_k(x,y)\big)^2 dydx \tag{7.17}$$

$$= \int_0^m \int_0^n \big(f(x,y)\big)^2 dydx - 2\sum_{k=1}^{t} u_k \int_0^m \int_0^n f(x,y)S_k(x,y)dydx$$

$$+ \sum_{k=1}^{t}\sum_{l=1}^{t} u_k u_l \int_0^m \int_0^n S_k(x,y)S_l(x,y)dydx \tag{7.18}$$

subject to $\boldsymbol{u} \geq \boldsymbol{0}$. Note that it is sufficient to minimize the square of the norm. Let us define

$$\begin{aligned} c_k &:= -2\int_0^m \int_0^n f(x,y)S_k(x,y)dydx \quad \text{and} \\ d_{k\ell} &:= 2\int_0^m \int_0^n S_k(x,y)S_\ell(x,y)dydx. \end{aligned}$$

Note that the numbers $d_{k\ell}$ need to be computed only once and then can be used for all new instances of the segmentation problem. We put $D = (d_{k\ell})_{k,\ell\in\{1,\ldots,t\}}$ and $\boldsymbol{c} := (c_k)_{k\in\{1,\ldots,t\}}$. As $\int_0^m \int_0^n (f(x,y))^2 dxdy$ is a constant, the minimization problem is equivalent to:

$$h(\boldsymbol{u}) = \frac{1}{2}\sum_{k=1}^{t}\sum_{\ell=1}^{t} d_{k\ell}u_k u_\ell + \sum_{k=1}^{t} c_k u_k \to \min \tag{7.19}$$

subject to $\boldsymbol{u} \geq \boldsymbol{0}$. Note that this problem is similar to the one in Section 6.3 and, as it is there, we have a convex quadratic programming problem with nonnegativity constraints. In matrix notation, we have to minimize

$$h(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T D\boldsymbol{u} + \boldsymbol{c}^T\boldsymbol{u} \tag{7.20}$$

subject to $\boldsymbol{u} \geq \boldsymbol{0}$. The dimension of the matrix D grows linearly with the number of segments in $\mathcal{S}'$. As the set of segments that can be realized by the MLC is large, we restrict ourselves to a subset. Thus, the most difficult problem is to find a set $\mathcal{S}'$, which is not too large, but enables a good quality of approximation. As mentioned in the previous section, we discretize the irradiation field in the horizontal direction and require the left and right leaf positions ℓ_i and r_i in row i to be integers in $[0, n]$. Therefore, using the concept of additivity, we can leave out larger segments that can be written as the sum of smaller ones. Thus, each segment is a sum of basic segments and

it is sufficient to put the mn basic segments into $\mathcal{S}'$, i.e. $\mathcal{S}' = \{S_{[i-1,i]\times[j-1,j]} \mid (i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\}\}$. Having computed the solution of the quadratic optimization step, one gets a coefficient for each basic segment, i.e. a real-valued matrix A. In clinical practice, one can adjust the irradiation time with a certain accuracy of $\frac{1}{k}$ monitor units for some $k > 1$. Thus, the matrix A can be multiplied by this parameter, rounded to integer values and decomposed with a discrete decomposition algorithm. The resulting coefficients of the segments should be divided by k and need not be integers if $k > 1$. The discrete decomposition algorithm can be chosen with regard to the technical and case-dependent requirements. Basically, every discrete decomposition algorithm is applicable.

Note that our choice of the set $\mathcal{S}'$ makes it easy to compute the matrix D. On the one hand, the nonzero region of $S_{[i-1,i]\times[j-1,j]}$ is $[i-1-\frac{\gamma}{2}, i+\frac{\gamma}{2}] \times [j-1-\frac{\gamma}{2}, j+\frac{\gamma}{2}]$ and therefore, many entries of D are zero, because the nonzero regions of the basic segments do not overlap. On the other hand, if the integral is not 0, it only depends on the overlap of the two nonzero regions and not on the position in the field. By using some appropriate case distinctions, one can compute the integrals analytically and avoid time consuming numerical integration. Table 7.1 shows the integrals $\int_0^m \int_0^n S_{[i-1,i]\times[j-1,j]}(x,y) S_{[k-1,k]\times[\ell-1,\ell]}(x,y) dxdy$ for linear and quadratic decline with $\gamma = 2$.

	linear decline	quadratic decline
$\int_0^m \int_0^n S_{[i-1,i]\times[j-1,j]}(x,y) S_{[k-1,k]\times[\ell-1,\ell]}(x,y) dydx$		
$\lvert i-k\rvert > 2$ or $\lvert j-\ell\rvert > 2$	0	0
$\lvert i-k\rvert = 2$, $\lvert j-\ell\rvert = 2$	$\frac{1}{576}$	$\frac{1}{14400}$
$(\lvert i-k\rvert = 2, \lvert j-\ell\rvert = 1)$ or $(\lvert i-k\rvert = 1, \lvert j-\ell\rvert = 2)$	$\frac{1}{96}$	$\frac{13}{7200}$
$(\lvert i-k\rvert = 2, \lvert j-\ell\rvert = 0)$ or $(\lvert i-k\rvert = 0, \lvert j-\ell\rvert = 2)$	$\frac{5}{288}$	$\frac{11}{2400}$
$\lvert i-k\rvert = 1$, $\lvert j-\ell\rvert = 1$	$\frac{1}{16}$	$\frac{169}{3600}$
$(\lvert i-k\rvert = 1, \lvert j-\ell\rvert = 0)$ or $(\lvert i-k\rvert = 0, \lvert j-\ell\rvert = 1)$	$\frac{5}{48}$	$\frac{143}{1200}$
$i = k$, $j = \ell$	$\frac{25}{144}$	$\frac{121}{400}$

Tab. 7.1: Values of $\int_0^m \int_0^n S_{[i-1,i]\times[j-1,j]}(x,y) S_{[k-1,k]\times[\ell-1,\ell]}(x,y) dydx$ for the different cases of the overlap of the basic segments $S_{[i-1,i]\times[j-1,j]}$ and $S_{[k-1,k]\times[\ell-1,\ell]}$ for linear and quadratic decline and $\gamma = 2$.

7.3 Summary of the approach

We briefly summarize the different steps of our approach:

1. Decide for a decline function t and a decline width $\gamma > 0$.
2. Model the basic fluence distributions $S_{[i-1,i]\times[j-1,j]} : [0,m] \times [0,n] \to [0,1]$ according to t and γ as described in Subsections 7.1.1 and 7.1.2.
3. Choose $\mathcal{S}' = \{S_{[i-1,i]\times[j-1,j]} \mid (i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}\}$.
4. Solve the problem **Approx-MIN-TC-Continuous** for $\mathcal{S}'$:

$$\Big\|f - \sum_{S\in\mathcal{S}'} u_S S\Big\|_2^2 = \Big\|f - \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} S_{[i-1,i]\times[j-1,j]}\Big\|_2^2 \to \min$$

   subject to $a_{ij} \geq 0$ for all $(i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}$ and put $A := (a_{ij})$. The value a_{ij} determines how often we need the basic fluence distribution $S_{[i-1,i]\times[j-1,j]}$.
5. Multiply A by a parameter $k > 1$, where k depends on the exactness in monitor units that is deliverable by the MLC. For example, $k = 10$ is applicable.
6. Round the values of A such that A becomes integral. In order to minimize the underdosage and overdosage effects due to the rounding, we consider the target fluence and the approximated fluence at the middle points $(x,y) = (i - 1/2, j - 1/2)$ of the bixel. We round up if $f(x,y) \geq \sum_{k=1}^{m}\sum_{\ell=1}^{n} a_{k\ell} S_{[k-1,k]\times[\ell-1,\ell]}(x,y)$ and round down if $f(x,y) < \sum_{k=1}^{m}\sum_{\ell=1}^{n} a_{k\ell} S_{[k-1,k]\times[\ell-1,\ell]}(x,y)$.
7. Decompose A into a sequence of leaf positions using a discrete decomposition algorithm. Choose the discrete approach depending on the type of constraints being considered. Note that we use the linearity of fluence distributions here!
8. Divide the computed coefficients of the segments by k.
9. Output the segmentation to the further treatment planning process. The segmentation has coefficients with an exactness of $\frac{1}{k}$ monitor units.

One point that needs to be discussed is that if we consider a bixel (i,j) with either $i \in \{1,\dots,\lceil\frac{\gamma}{2}\rceil\} \cup \{m - \lceil\frac{\gamma}{2}\rceil + 1,\dots,m\}$ or $j \in \{1,\dots,\lceil\frac{\gamma}{2}\rceil\} \cup \{n - \lceil\frac{\gamma}{2}\rceil + 1,\dots,n\}$, then our definition of basic fluence distributions would lead to the effect that radiation is transmitted to regions outside from $[0,m] \times [0,n]$. We solve this problem by increasing the considered area, i.e. work with $[-\frac{\gamma}{2}, m + \frac{\gamma}{2}] \times [-\frac{\gamma}{2}, n + \frac{\gamma}{2}]$ and set the target fluence to 0 in this added area. As candidates for the optimization, still only bixels (i,j) with $1 \leq i \leq m$ and $1 \leq j \leq n$ are allowed.
One might ask why the continuous segment model is not integrated into the treatment planning process before the segmentation step such that the intensity matrices coming to the segmentation step already contain the "correct" fluence information according to the new model. But this is not a good idea because the computational complexity of this part of the planning would increase (cf. [25] for details). Including the model into

the segmentation step can be handled efficiently. However, the new model is used in the post-optimization steps to improve dose calculation for the chosen segments (cf. again [25]).

7.4 Results

We analyze how good the quality of the approximation in the quadratic optimization step is and solve the quadratic programming problem with a self-designed Projected-Newton routine that was implemented in C++. We again use the 475 clinical intensity matrices (below denoted by $\tilde{A}$) that were provided by the Department of Radiotherapy and Radio-Oncology at the University Medical Center Hamburg-Eppendorf. They were computed for discrete segmentation and we used bilinear interpolation to transform them into target fluence distributions f. For the basic fluence distributions, we use a decline width of $\gamma = 2$ and linear decline as well as quadratic decline.
Table 7.2 shows the approximation errors for the continuous approximation approach. We also compare our continuous decomposition method with common discrete segmentation methods. In our model of fluence distributions, a discrete segmentation of $\tilde{A}$ corresponds to a fluence containing $\tilde{a}_{ij}$ times the basic segment $S_{[i-1,i]\times[j-1,j]}$. Thus, assuming that radiation behaves according to our new model, we calculate the fluence distribution errors of discrete and continuous segmentation. To be exact, we compare the following two settings:

- Continuous segmentation: The continuous segmentation approach is used and the computed matrix A (resulting from the quadratic optimization step and a rounding procedure with $k = 10$) is decomposed with a discrete decomposition method. This approach leads to the fluence

$$f_{con}(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} S_{[i-1,i]\times[j-1,j]}(x,y).$$

- Discrete Segmentation: The matrix $\tilde{A}$ is decomposed with a discrete decomposition method. This yields a fluence of

$$f_{discr}(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} \tilde{a}_{ij} S_{[i-1,i]\times[j-1,j]}(x,y).$$

According to our approach, the fluence f_{con} is the optimal approximation of f that can be decomposed into a weighted sum of basic segments.
The objective value of the quadratic programming problem $\|f - f_{con}\|_2^2$ measures the quality of the approximation. As the value $\|f\|_2^2 = \int_0^m \int_0^n (f(x,y))^2 dydx$ differs for the different instances, a comparison of the relative approximation errors $(\|f - f_{con}\|_2^2) \cdot (\|f\|_2^2)^{-1}$ makes sense. Analogously, we report the values of $\|f - f_{discr}\|_2^2$ and $(\|f - f_{discr}\|_2^2) \cdot (\|f\|_2^2)^{-1}$. The approximation errors for the continuous approach are overall small, but the quadratic decline function performs even better than the

linear one. For almost all instances, the relative error is less than 1% and thus, a good approximation of the target fluence is possible. The discrete decomposition yields larger approximation errors. On average, the errors are about one quarter larger than for continuous segmentation with linear decline and about 66% larger than for continuous segmentation with quadratic decline. It took approximately one hour of computation time (on a 2.5GHz workstation) for each of the two decline functions to produce the results for all instances.

	linear decline		
	average	$\frac{\|f-f_{con}\|_2^2}{\|f\|_2^2}$ min.	$\frac{\|f-f_{con}\|_2^2}{\|f\|_2^2}$ max.
m	19.47	24	22
n	20.76	22	19
$\|f\|_2^2$	31766.37	459437	1861.04
$\|f - f_{con}\|_2^2$	624.70	3026.21	234.14
$(\|f - f_{con}\|_2^2) \cdot (\|f\|_2^2)^{-1}$	0.04	0.0066	0.13
$\|f - f_{discr}\|_2^2$	775.65	3687.94	269.23
$(\|f - f_{discr}\|_2^2) \cdot (\|f\|_2^2)^{-1}$	0.05	0.008	0.14
	quadratic decline		
m	19.47	20	22
n	20.76	20	19
$\|f\|_2^2$	31766.37	175200.32	1861.04
$\|f - f_{con}\|_2^2$	77.36	158.07	32.69
$(\|f - f_{con}\|_2^2) \cdot (\|f\|_2^2)^{-1}$	0.005	0.0009	0.0176
$\|f - f_{discr}\|_2^2$	128.50	241.04	48.22
$(\|f - f_{discr}\|_2^2) \cdot (\|f\|_2^2)^{-1}$	0.0084	0.0014	0.026

Tab. 7.2: Numerical results for Approx-MIN-TC-Continuous for linear and quadratic decline with decline width $\gamma = 2$. The three columns provide the average results and the results for the instance with the minimum and maximum relative approximation error.

Figure 7.7 shows an example of a target fluence distribution f and its best approximation $\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}S_{[i-1,i]\times[j-1,j]}$ with quadratic decline and a decline width of $\gamma = 2$. Our results show, that our continuous segmentation approach improves the approximation of the target fluences in comparison to discrete decompositions and thus, a more accurate delivery is possible.
Finally, we chose one of the clinical matrix instances with $\|f\|_2^2 = 10408.40$ and computed $\|f - f_{con}\|_2^2$ and $\|f - f_{discr}\|_2^2$ for various values of the decline width γ. The relative approximation errors are illustrated in Figure 7.8. As expected, the differences between the errors of continuous and discrete segmentation increase with increasing γ. Thus, the quadratic optimization step produces matrices for decomposition that correspond to fluence distributions closer to the target fluence. The more realistic model of the characteristics of radiation leads to more exact matrices that have to be decomposed with a discrete method. The hope is that the quality of treatment plans can be improved with this approach.

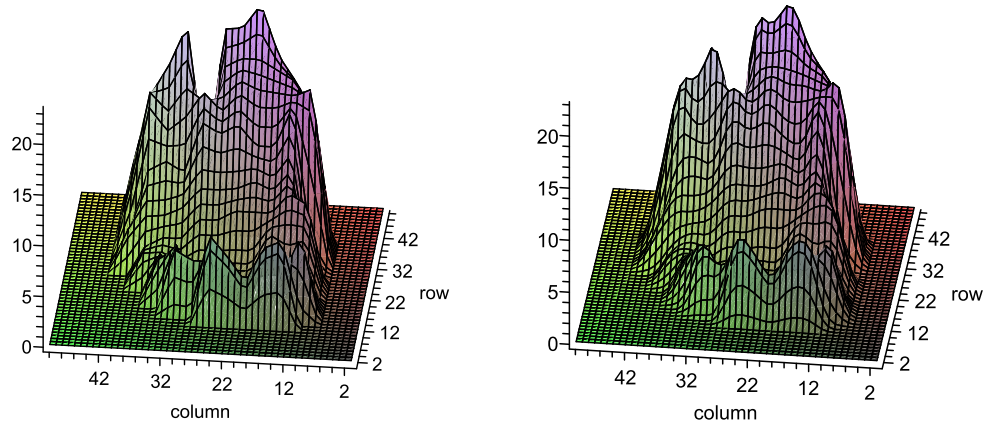

Fig. 7.7: A target fluence distribution on the left and its approximation with quadratic decline and a decline width of $\gamma = 2$ on the right.

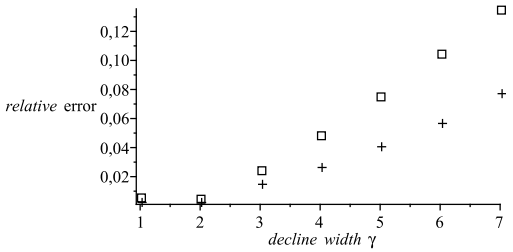

Fig. 7.8: Relative approximation errors $\|f - f_{con}\|_2^2 \cdot (\|f\|_2^2)^{-1}$ (lower curve with +-signs) and $\|f - f_{discr}\|_2^2 \cdot (\|f\|_2^2)^{-1}$ (upper curve with box-signs) for one clinical instance with $\|f\|_2^2 = 10408.40$ and various values of the decline width γ.

As testing on clinical cases is a complex and time-consuming procedure, we only did a single test on the same case as in Section 6.4.2. The discrete decomposition algorithm used is the heuristic segmentation algorithm presented in Section 6.4.1 (for segmentations into large segments) with the parameters $f = w = 4$ and $h = 2$, which perform well in practice. The result is that the dose volume histograms were equivalent for discrete segmentation and continuous segmentation both for linear and quadratic decline (cf. Figure 7.9). But comparing the continuous segmentation with quadratic decline and the discrete segmentation, the number of segments and the monitor units could be reduced further (45 segments instead of 51 and 19200 monitor units instead of 21865). Additionally, the homogeneity of the 95% isodose was improved using the continuous approach. The basic outcome is, that with our sophisticated methods we are close to treatment plans where no more significant improvement is possible, which is satisfying. But slight improvements can be achieved by the continuous method because of the more exact modeling of the border of the radiation field. Of course, for substantial results on the continuous method, further research is necessary.

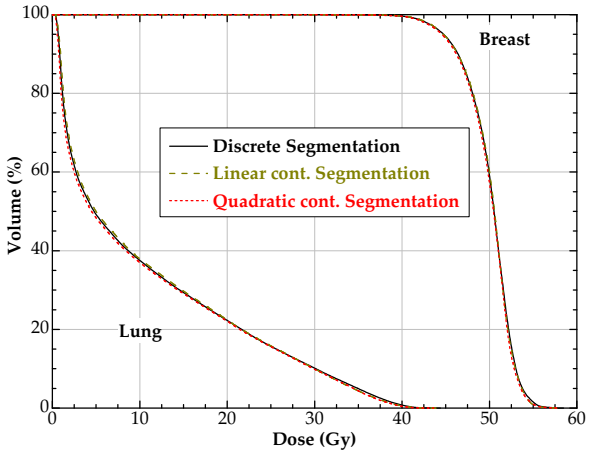

Fig. 7.9: Dose volume histogram for discrete and continuous segmentation.

7.5 A column generation approach to generate segments

Using the additive model of segments with the decline functions, the approach from the previous section is probably the most useful one. If we want to fit our model even closer to reality, we have to take into account, that a small part of radiation is transmitted through the leaves of the MLC. This part is called leakage radiation. Let therefore the leakage factor $\ell \in \mathbb{R}_+$ be given. The value of ℓ depends on the energy of the radiation and can be assumed to be in $[0.02, 0.05]$. Taking the leakage value into account leads to the consequence, that the segments no longer have the additivity property (7.2). The reason is, that for disjoint uncovered regions O and O', if $S_O(x, y) = \ell$ and $S_{O'}(x, y) = \ell$, then $S_{O \cup O'}(x, y)$ should be also equal to ℓ which contradicts the additivity property. Thus, the approach from the previous section using the set of basic segments for the QP-approach is no longer applicable and we cannot build segments with larger uncovered regions by summing up segments with smaller ones.

A possible model of the segments that can be used is the following. We take the segment model from the previous section and redefine the fluence distribution for each uncovered region $O \subseteq [0, m] \times [0, n]$ by

$$S_O(x, y) := \max(\ell, S_O(x, y)). \tag{7.21}$$

Of course, other extensions are possible.

Thus, we have to work with fluence distributions of final segments and cannot use just basic segments for the decomposition. As in the previous section, let the set of allowed segments $\mathcal{S}'$ be the set of segments corresponding to integral leaf positions. If we take all possible segments into account, the approximation problem has too many variables. Therefore, we use column generation as in Section 6.3.

Let D, $\boldsymbol{c}$, $D_{\mathcal{S}''}$ and $\boldsymbol{c}_{\mathcal{S}''}$ be defined as in Section 6.3, with the exception that the entries are continuous integrals instead of sums now. As it is there, our aim is to find a solution for the master problem

$$\min \frac{1}{2} \boldsymbol{u}^T D \boldsymbol{u} + \boldsymbol{c}^T \boldsymbol{u} \quad \text{subject to} \quad \boldsymbol{u} \geq \boldsymbol{0}. \tag{7.22}$$

For an optimal solution $\boldsymbol{u}$ of the master problem, the Karush-Kuhn-Tucker (KKT) conditions (6.25), (6.26), (6.27) and (6.28) are fulfilled.
If we have a solution for the master problem with only a limited number of allowed segments $\mathcal{S}'' \subseteq \mathcal{S}'$, we have to find a segment $S \in \mathcal{S}' \setminus \mathcal{S}''$ in the subproblem that minimizes

$$\sum_{S' \in \mathcal{S}''} d_{S,S'} u_{S'} + c_S \geq 0$$

in order to improve the objective function of the master problem.
Representing a segment S by the values in the bixels (i,j) and using the notations

$$d_{i,j}^{S,S'} := 2 \int_{i-1}^{i} \int_{j-1}^{j} S(x,y) S'(x,y) dx dy$$

and

$$c_{i,j}^{S} := -2 \int_{i-1}^{i} \int_{j-1}^{j} S(x,y) f(x,y) dx dy,$$

we get the subproblem

$$\min_{S \in \mathcal{S}'} \sum_{S' \in \mathcal{S}''} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(d_{i,j}^{S,S'} u_{S'} + c_{i,j}^{S} \right). \tag{7.23}$$

After regrouping the terms, we have to find a segment S that minimizes

$$\min_{S \in \mathcal{S}'} \sum_{i=1}^{m} \sum_{j=1}^{n} \Big(\sum_{S' \in \mathcal{S}''} d_{i,j}^{S,S'} u_{S'} + c_{i,j}^{S} \Big). \tag{7.24}$$

This subproblem can be solved using a graph theoretical approach. The basic idea is, that for each bixel (i,j) we have to compute its contribution $\sum_{S' \in \mathcal{S}''} d_{i,j}^{S,S'} u_{S'} + c_{i,j}^{S}$ to the objective function.
Let a shape be given by the sequence of its leaf positions $((\ell_1, r_1), \ldots, (\ell_m, r_m))$. Thus, a bixel (i,j) is open, iff $\ell_i \leq j \leq r_i$. Remember, that the leafs are situated in the middle of the decline region. Thus, if we use a decline width of γ, an open bixel (i,j) transmits radiation to the bixels (i',j') with $|i-i'| \leq \lceil \gamma/2 \rceil$ and $|j-j'| \leq \lceil \gamma/2 \rceil$. In clinical practice, the penumbra actually has a width of 0.5cm - 1cm and thus, we will restrict ourselves to the case $\gamma \leq 2$ here. In this case, the value of a segment at a position $(x,y) \in [i-1,i] \times [j-1,j]$ in bixel (i,j) only depends on whether the bixels (k,ℓ) for $k \in \{i-1, i, i+1\}$ and $\ell \in \{j-1, j, j+1\}$ are open or closed.
Concluding, for $\gamma \leq 2$ and an arbitrary segment, if we know the leaf positions (ℓ_{i-1}, r_{i-1}), (ℓ_i, r_i) and (ℓ_{i+1}, r_{i+1}) we know everything about the segment values in row i. For $S' \in \mathcal{S}''$ and a new segment S, the values of the quantities $d_{i,j}^{S,S'} u_{S'}$ and $c_{i,j}^{S}$ for $j \in [n]$ do only depend on the leaf positions in rows $i-1, i$ and $i+1$. Thus, we can solve the subproblem by shortest path computation in the following acyclic digraph $G = (V, E)$:

$$
\begin{aligned}
V &= \{D, D'\} \cup \left\{ (i,\ell,r,\ell',r') \;\middle|\; \begin{smallmatrix} 1\le i\le m-1, \\ 1\le \ell\le r\le n \text{ or } (\ell,r)=(1,0), \\ 1\le \ell'\le r'\le n \text{ or } (\ell',r')=(1,0) \end{smallmatrix} \right\} \cup \\
&\quad \left\{ (0,1,0,\ell,r), (m,\ell,r,1,0) \;\middle|\; \begin{smallmatrix} 1\le \ell\le r\le n \text{ or} \\ (\ell,r)=(1,0) \end{smallmatrix} \right\}, \\
E &= E_1 \cup E_2 \cup E_3 \text{ with} \\
E_1 &= \{(D, (0,1,0,\ell,r)) \mid (0,1,0,\ell,r) \in V\}, \\
E_2 &= \{((m,\ell,r,1,0), D') \mid (m,\ell,r,1,0) \in V\}, \\
E_3 &= \left\{ ((i-1,\ell,r,\ell',r'), (i,\ell',r',\ell'',r'')) \;\middle|\; \begin{smallmatrix} (i-1,\ell,r,\ell',r')\in V, \\ (i,\ell',r',\ell'',r'')\in V \end{smallmatrix} \right\}.
\end{aligned}
$$

Each vertex of type (i,ℓ,r,ℓ',r') corresponds to the leaf positions (ℓ,r) in row i and to the leaf positions (ℓ',r') in row $i+1$, respectively. Traversing an edge of type $((i-1,\ell,r,\ell',r'),(i,\ell',r',\ell'',r''))$ represents the choice of the leaf positions (ℓ,r) in row $i-1$, (ℓ',r') in row i and (ℓ'',r'') in row $i+1$. Now we define a weight function $c : E \to \mathbb{R}$ by

$$
\begin{aligned}
c(e) &= \sum_{j=1}^{n} \Big(\sum_{S'\in\mathcal{S}''} d_{i,j}^{S,S'} u_{S'} + c_{i,j}^{S} \Big) \text{ for all } e \in E_3 \\
c(e) &= 0 \text{ for all } e \in E \setminus E_3.
\end{aligned}
$$

Note, that the edge weights do only depend on the information given by the start and the end node of the edge, namely on $\ell, r, \ell', r', \ell''$ and r''. A shortest path in the graph $G = (V, E)$ defined above corresponds to a segment S that minimizes (7.24). If the length of the shortest path is 0 and all leaf positions are $(1,0)$, there is no segment that violates the KKT conditions and our current solution of the master problem is indeed globally optimal. Otherwise, the computed segment S is likely capable of improving the objective value of our master problem and thus is added to $\mathcal{S}''$. The solving of the subproblem has to be changed adequately if the segments shall satisfy some given constraints.

8. SUMMARY AND OPEN QUESTIONS

In this thesis, we presented a comprehensive collection of algorithms for the segmentation step in IMRT planning. Whereas the results on exact TG-segmentations, on approximation for DT-minimization with ICC or on approximation for TC-minimization using an arbitrarily given set of segments are of theoretical interest, the other approximative approaches aim at modeling the needs of clinical practice. Both the reduction of the delivery time and the realization of fluence distributions that are close to the target fluence, but also decomposable into segments with good dosimetric properties, are of high clinical relevance. The most realistic model is probably the continuous one as it takes the physical characteristics of radiation into account. First experiments show the potential of our approaches, as improved treatment plans could be provided for a clinical case. Optimal parameter fitting and further adjustment of the heuristic optimization steps might enable further enhancements in the near future. A challenging task is the incorporation of the transmission coefficient into the continuous segmentation model without using the time-consuming column-generation approach.
From a mathematical point of view, the question whether there exists a combinatorial polynomial time algorithm for finding general TG-segmentations remains open. Furthermore, the tightness of the (in)approximability results developed in Section 6.1 should be object of future research. It is also interesting to consider the problems using different ℓ_p-norms.

BIBLIOGRAPHY

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

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[4] D. Baatar, N. Boland, S. Brand, and P.J. Stuckey. Minimum cardinality matrix decomposition into consecutive-ones matrices: CP and IP approaches. *LNCS*, 4510:1–15, 2007.

[5] D. Baatar, N. Boland, R. Johnston, and H.W. Hamacher. A new sequential extraction heuristic for optimizing the delivery of cancer radiation treatment using multileaf collimators. *Informs Journal on Computing*, 21(2):224–241, 2009.

[6] D. Baatar, H.W. Hamacher, M. Ehrgott, and G.J. Woeginger. Decomposition of integer matrices and multileaf collimator sequencing. *Discrete Appl. Math.*, 152(1-3):6–34, 2005.

[7] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1996.

[8] J.L. Bedford and S. Webb. Constrained segment shapes in direct-aperture optimization for step-and-shoot IMRT. *Med. Phys.*, 33(4):944–958, 2006.

[9] 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.

[10] T. Bortfeld. IMRT: A review and preview. *Phys. Med. Biol.*, 51:R363–R379, 2006.

[11] 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.

[12] A.L. Boyer and C.Y. Yu. Intensity–modulated radiation therapy with dynamic multileaf collimators. *Semin. Radiat. Oncol.*, 9:48–59, 1999.

[13] J. Cardinal, S. Fiorini, and G. Joret. Tight results on minimum entropy set cover. *Algorithmica*, 51(1), 2008.

[14] F. Carlsson. Combining segment generation with direct step-and-shoot optimization in intensity-modulated radiation therapy. Report TRITA-MAT-2007-OS3, Department of Mathematics, Royal Institute of Technology, Stockholm, 2007.

[15] F. Carlsson and A. Forsgren. A conjugate-gradient based approach for approximate solutions of quadratic programs. Report TRITA-MAT-2008-OS2, Department of Mathematics, Royal Institute of Technology, Stockholm, 2008.

[16] D.Z. Chen, K. Engel, and C. Wang. A new algorithm for a field splitting problem in intensity-modulated radiation therapy. *Algorithmica*, DOI 10.1007/s00453-010-9429-6, 2010.

[17] D.Z. Chen, X. Hu, S. Luan, S.A. Naqvi, C. Wang, and C.X. Yu. Generalized geometric approaches for leaf sequencing problems in radiation therapy. *LNCS*, 3341:271–281, 2005.

[18] D.Z. Chen, X.S. Hu, C. Wang, S. Luan, and X. Wu. Mountain reduction, block matching, and applications in intensity-modulated radiation therapy. *Int. J. Comput. Geometry Appl.*, 18(1/2):63–106, 2008.

[19] D.J. Convery and M.E. Rosenbloom. The generation of intensity–modulated fields for conformal radiotherapy by dynamic collimation. *Phys. Med. Biol.*, 37(6):1359–1374, 1992.

[20] 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.

[21] 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.

[22] B. Doerr and M. Wahlström. Randomized rounding in the presence of a cardinality constraint. In *Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX). SIAM*, Philadelphia: 162–174, 2009.

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

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

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

[26] K. Engel and A. Kiesel. Approximated matrix decomposition for IMRT planning with multileaf collimators. *OR Spectrum*, DOI 10.1007/s00291-009-0168-5, 2009.

[27] K. Engel and E. Tabbert. Fast simultaneous angle, wedge, and beam intensity optimization in inverse radiotherapy planning. *Optimization and Engineering*, 6(4):393–419, 2005.

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

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

[30] C. Engelbeen and A. Kiesel. Binary matrix decompositions without tongue-and-groove underdosage for radiation therapy planning. to appear in *Algorithmic Operations Research*, 2010.

[31] A.T. Ernst, V.H. Mak, and L.R. Mason. An exact method for the minimum cardinality problem in the treatment planning of intensity-modulated radiotherapy. *Informs Journal on Computing*, 21(4):562–574, 2009.

[32] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[33] U. Feige, L. Lovász, and P. Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.

[34] M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP–completeness*. W.H. Freeman, New York, 1979.

[35] T. Gauer, D. Albers, F. Cremers, R. Harmansa, R. Pellegrini, and R. Schmidt. Design of a computer-controlled multileaf collimator for advanced electron radiotherapy. *Phys. Med. Biol.*, 53:5987–6003, 2006.

[36] T. Gauer, K. Engel, A. Kiesel, D. Albers, and D. Rades. Comparison of electron IMRT to helical photon IMRT and conventional photon irradiation for treatment of breast and chest wall tumours. *Radiother. Oncol.*, 94:313–318, 2010.

[37] T. Gauer, J. Sokoll, F. Cremers, R. Harmansa, M. Luzzara, and R. Schmidt. Characterization of an add-on multileaf collimator for electron beam therapy. *Phys. Med. Biol.*, 53:1071–1085, 2008.

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

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

[40] T. Kalinowski. Realization of intensity modulated radiation fields using multileaf collimators. *LNCS*, 4123:995–1040, 2006.

[41] T. Kalinowski. Multileaf collimator shape matrix decomposition. In *Optimization in Medicine and Biology. G.J. Lim and E.K.Lee*, Auerbach Publishing: 253–286, 2008.

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

[43] T. Kalinowski. The complexity of minimizing the number of shape matrices subject to minimal beam-on time in multileaf collimator field decomposition with bounded fluence. *Discrete Appl. Math.*, 157:2089–2104, 2009.

[44] T. Kalinowski. A min cost network flow formulation for approximated MLC segmentation. *Networks*, DOI 10.1002/net.20394, 2010.

[45] T. Kalinowski and A. Kiesel. Approximated MLC shape matrix decomposition with interleaf collision constraint. *Algorithmic Operations Research*, 4(1):49–57, 2009.

[46] P. Källman, B. Lind, A. Eklöf, and A. Brahme. Shaping of arbitrary dose distributions by dynamic multileaf collimation. *Phys. Med. Biol.*, 33:1291–1300, 1988.

[47] 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.

[48] 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.

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

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

[51] A. Kiesel. A function approximation approach to the segmentation step in IMRT planning. *OR Spectrum*, DOI:10.1007/s00291-009-0187-2, 2009.

[52] A. Kiesel. Constrained approximate multileaf collimator field segmentation for IMRT with electrons. submitted to *Operations Research*, 2010.

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

[54] J. Lim, M.C. Ferris, S.J. Wright, D.M. Shepard, and M.A. Earl. An optimization framework for conformal radiation treatment planning. *Informs Journal on Computing*, 19(3):366–380, 2007.

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

[56] 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.

[57] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post-quantum Cryptography. D.J. Bernstein and J. Buchmann*, Springer, 2008.

[58] R. Motwani, J. Naor, and P. Raghavan. Randomized approximation algorithms in combinatorial optimization. In *Approximation Algorithms for NP-hard Problems. Hochbaum, D.*, PWS Publishing Co., Boston: 447–481, 1997.

[59] Gurobi Optimization. Gurobi Optimizer 3.0, www.gurobi.com.

[60] W. Que. Comparison of algorithms for multileaf collimator field segmentation. *Med. Phys.*, 26:2390–2396, 1999.

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

[62] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *J. Comput. Syst. Sci.*, 370:130–143, 1988.

[63] H.E. Romeijn, R.K. Ahuja, J.F. Dempsey, and A. Kumar. A column generation approach to radiation therapy treatment planning using aperture modulation. *SIAM Journal on Optimization*, 15(3):838–862, 2005.

[64] C.G. Rowbottom, V.S. Khoo, and S.W. Webb. Simultaneous optimization of beam orientations and beam weights in conformal radiotherapy. *Med. Phys.*, 28:1696–1702, 2001.

[65] 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.

[66] D.M. Shepard, M.A. Earl, X.A. Li, S. Naqvi, and C. Yu. Direct aperture optimization: A turnkey solution for step-and-shoot IMRT. *Med. Phys.*, 29(6):1007–1018, 2002.

[67] P. Spellucci. *Numerische Verfahren der nichtlinearen Optimierung.* Birkhäuser, 1993.

[68] S.V. Spirou and C.S. Chui. Generation of arbitrary intensity profiles by dynamic jaws or multileaf collimators. *Med. Phys.*, 21:1031–1041, 1994.

[69] A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *42nd IEEE Symposium on Foundations of Computer Science, Las Vegas*. IEEE Computer Soc., Los Alamitos: 588–597, 2001.

[70] X. Wu and Y. Zhu. A global optimization method for three–dimensional conformal radiotherapy treatment planning. *Phys. Med. Biol.*, 46:107–119, 2001.

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

APPENDIX

Pseudocodes of the algorithms from Section 5.1:

Algorithm 9 Computation of the min-max sequence

Input: vectors $\boldsymbol{a}$, $\underline{\boldsymbol{a}}$ and $\overline{\boldsymbol{a}}$.

Put $\boldsymbol{s} = (0, 0)$ and $j = 1$.
while $j \leq n$ **do**
Put $j_1 = j_2 = j$ and $max = \underline{a}_j$.
while $\overline{a}_j \geq max$ and $j \leq n$ **do**
if $\underline{a}_j = max$ **then**
$j_2 = j$;
else if $\underline{a}_j > max$ **then**
$j_1 = j_2 = j$ and $max = \underline{a}_j$;
end if
$j = j + 1$;
end while
Add j_1 and j_2 to the sequence $\boldsymbol{s}$.
Put $j_1 = j_2 = j$ and $min = \overline{a}_j$.
while $\underline{a}_j \leq min$ and $j \leq n$ **do**
if $\overline{a}_j = min$ **then**
$j_2 = j$;
else if $\overline{a}_j < min$ **then**
$j_1 = j_2 = j$ and $min = \overline{a}_j$;
end if
$j = j + 1$;
end while
if $j \leq n$ **then**
Add j_1 and j_2 to the sequence $\boldsymbol{s}$.
end if
end while
Add $n + 1$ twice to the sequence $\boldsymbol{s}$.

Output: min-max-sequence $\boldsymbol{s}$.

Algorithm 10 Solution of the Problem **CDTMTC-Row**

Input: vectors $\boldsymbol{a}$, $\underline{\boldsymbol{a}}$ and $\overline{\boldsymbol{a}}$, bound C with $c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(\boldsymbol{a}) \le C < c(\boldsymbol{a})$.

Initialize $p_{i,j,k} = q_{i,j,k} = \infty$ for all i, j, k.
for $i = -(a_1 - \underline{a}_1)$ to 0 **do**
  $p_{i,1,-i} = -i$;
  $q_{i,1,-i} = 0$;
end for
for $j = 1$ to $n-1$ **do**
  for $i_1 = -(a_j - \underline{a}_j)$ to $(\overline{a}_j - a_j)$ **do**
    for $i_2 = -(a_{j+1} - \underline{a}_{j+1})$ to $\min\{i_1 + (a_j - a_{j+1})_+, \overline{a}_{j+1} - a_{j+1}\}$ **do**
      $d = (a_{j+1} - a_j)_+ - (a_{j+1} - a_j + i_2 - i_1)_+$;
      for $k_1 = 0$ to $c(j) - c_{\underline{\boldsymbol{a}},\overline{\boldsymbol{a}}}(j)$ **do**
        $k_2 = k_1 + d$;
        if $p_{i_1,j,k_1} + |i_2| < p_{i_2,j+1,k_2}$ **then**
          $p_{i_2,j+1,k_2} = p_{i_1,j,k_1} + |i_2|$;
          $q_{i_2,j+1,k_2} = i_1$;
        end if
      end for
    end for
  end for
end for
$z = \infty$;
$k = c(\boldsymbol{a}) - C$;
for $i = -(a_n - \underline{a}_n)$ to $(\overline{a}_n - a_n)$ **do**
  if $p_{i,n,k} < z$ **then**
    $z = p_{i,n,k}$;
    $b_n = a_n + i$;
  end if
end for
$i_1 = b_n - a_n$;
for $j = n-1$ downto 1 **do**
  $i_2 = i_1$;
  $i_1 = q_{i_2,j+1,k}$;
  $b_j = a_j + i_1$;
  $k = k + ((b_{j+1} - b_j)_+ - (a_{j+1} - a_j)_+)$;
end for

Output: vector $\boldsymbol{b}$.

Pseudocode of the algorithm from Section 6.2:

Algorithm 11 LOC-left

Input: vector $\boldsymbol{v}$.
  for $i = min$ to max **do**
    $tc_{i1} = |v_1 - i|$;
  end for
  $tc_{ij} = \infty$ for all $i \in [min, max], j > 1$
  for $j = 2$ to k **do**
    for $i = min$ to max **do**
      for $i' = min$ to i **do**
        if $tc_{i,j-1} + |v_j - i'| < tc_{i'j}$ **then**
          $tc_{i'j} = tc_{i,j-1} + |v_j - i'|$;
          $pre_{i'j} = i$;
        end if
      end for
    end for
  end for
  $opt = \min_{i \in [min,max]} tc_{i,k}$;
  Let i_{opt} be one of the indices with $tc_{i_{opt},k} = opt$.
  for $j = k$ downto 1 **do**
    $w_j = i_{opt}$;
    If still $j > 1$, then $i_{opt} = pre_{i_{opt},j}$.
  end for
Output: vector $\boldsymbol{w}$.

Pseudocode of the algorithm from Section 6.4:

Algorithm 12 Find L-Segment

Compute for each row $i \in [1, m-h+1]$ the set of useful rectangles $\mathcal{R}_i$.
$i_{start} = \min(\{k \in [m-f+1] \mid \boldsymbol{a}_k \not\leq \mathbf{0},\ \mathcal{R}_k \neq \emptyset\} \cup \{m+1\})$;
$i_{end} = \max(\{k \in [m-h+1] \mid \boldsymbol{a}_{k+h-1} \not\leq \mathbf{0},\ \mathcal{R}_k \neq \emptyset\} \cup \{0\})$;
if $i_{start} = m+1$ **then**
  Return $done = 1$.
end if
Segment $S = \mathbf{0}$;
while $\|A - S\|_1 \geq \|A\|_1$ **do**
  Segment $S = \mathbf{0}$;
  Choose $R \in \mathcal{R}_{i_{start}}$ with maximal value of $(\lambda p_R^+ + u_R)$ and add it to S.
  $i = i_{start} + 1$;
  while $r_{i-1} - \ell_{i-1} \geq w - 1$ and ($i \leq i_{end}$ or $i - 1 < i_{start} + f - 1$) **do**
    if $i \leq i_{end}$ and there is an S-feasible $R \in \mathcal{R}_i$ **then**
      Choose an S-feasible $R \in \mathcal{R}_i$ with maximal value of $(\lambda p_R^+ + u_R)$ and add it to S.
    else if $\boldsymbol{s}_i = \mathbf{0}$ and $i - 1 < i_{start} + f - 1$ **then**
      Put $\ell_i = \ell_{i-1}$ and $r_i = r_{i-1}$.
    end if
    $i = i + 1$;
  end while
  if $\|A - S\|_1 \geq \|A\|_1$ **then**
    $i_{start} = \min(\{k \in [i_{start}+1, m-f+1] \mid \boldsymbol{a}_k \not\leq \mathbf{0},\ \mathcal{R}_k \neq \emptyset\} \cup \{m+1\})$;
    if $i_{start} = m+1$ **then**
      Return $done = 1$.
    end if
  end if
end while
Return segment S.

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig 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.

Rostock, den 28. September 2010

Curriculum Vitae

Antje Kiesel

Personal data

Date of birth:	December 17th, 1983
Place of birth:	Bützow
Nationality:	German
Family status:	married, one child

Education

Since Jan 2008	Ph.D. student at the Institute for Mathematics, University of Rostock, Germany **Doctoral thesis:** Approximated multileaf collimator field segmentation **Supervisor:** Prof. Dr. Konrad Engel scholarship from the *Studienstiftung des Deutschen Volkes*
Oct 2003 - Dec 2007	Diploma studies in mathematics at the Institute for Mathematics, University of Rostock, Germany **Final Grade:** 1,0 (very good) **Diploma Thesis:** Approximated matrix decomposition for IMRT planning with multileaf collimators **Supervisor:** Prof. Dr. Konrad Engel scholarship from the *Studienstiftung des Deutschen Volkes*
Oct 2003 - Dec 2007	Diploma studies in business mathematics at the Institute for Mathematics, University of Rostock, Germany all diploma examinations passed

Work experience

Since Oct 2005	Teaching assistant at the Institute for Mathematics, University of Rostock, for different courses (Computer Algebra with Maple, Discrete Mathematics and Optimization, Probability Theory I, Linear Algebra I)
Sep 2009	Participation in the block course "Combinatorial Optimization at Work", FU Berlin
Aug 2007 - Sep 2007	Internship at the HSH Nordbank, Kiel
Mar 2007 - Jul 2007	Scientific assistant at the Institute for Computer Science, University of Rostock
Oct 2004 - Jul 2006	Teaching assistant at the Institute for Computer Science, University of Rostock, correction of exercises

Thesen zur Dissertation

Approximated multileaf collimator field segmentation

von Antje Kiesel

1. In der Strahlentherapiebehandlung werden Mehrlamellenkollimatoren zur Feldformung eingesetzt. Durch Überlagerung unterschiedlich geformter Felder soll eine möglichst exakte Realisierung der Zieldosis im Tumorgewebe erreicht werden, während umliegende Organe bestmöglich vor der Strahlung geschützt werden sollen. Die möglichen Lamellenpositionen des Kollimators (Segmente) werden durch 0-1-Matrizen, bei denen die Einsen in jeder Zeile aufeinanderfolgend sind, beschrieben. Die Menge aller Segmente sei mit $\mathcal{S}$ bezeichnet. Die Zielfluenz ist gegeben durch eine ganzzahlige nichtnegative $m \times n$-Matrix A, so dass das allgemeine exakte Segmentierungsproblem im Finden einer Zerlegung

$$A = \sum_{S \in \mathcal{S}} u_S S$$

mit ganzzahligen nichtnegativen Koeffizienten u_S besteht. Dabei wird der Wert $DT := \sum_{S \in \mathcal{S}} u_S$ als *Delivery Time* bezeichnet. Sie ist ein Maß für die Gesamtbestrahlungszeit des Patienten.

2. Aufgrund von physikalischen und dosimetrischen Gründen sind im klinischen Alltag allerdings eine Reihe von Nebenbedingungen zu beachten, so dass sich das exakte Segmentierungsproblem mit der Segmentmenge $\mathcal{S}$ als nicht praxistauglich erweist. Oftmals ist durch physikalische Einschränkungen des Gerätes bzw. durch ungünstige Eigenschaften bestimmter Feldformen nur eine Teilmenge $\mathcal{S}' \subseteq \mathcal{S}$ der Segmente zulässig. Zudem ist die Delivery Time oft unakzeptabel groß und soll aus Effizienzgründen sowie zum Schutz des Patienten reduziert werden. Deswegen lassen sich ein exaktes Segmentierungsproblem mit Nebenbedingungen sowie zwei Klassen von Approximationsproblemen formulieren, um diesen verschiedenen Anforderungen gerecht zu werden. Sei dazu $\|\cdot\|$ eine gegebene Vektornorm. Diese wird auch auf $m \times n$-Matrizen angewendet, indem wir diese als Vektoren der Länge mn auffassen.

- Gegeben sei eine Teilmenge $\mathcal{S}' \subseteq \mathcal{S}$, die exakte Zerlegungen aller Matrizen ermöglicht.

  MIN-DT: Finde eine Zerlegung $A = \sum_{S \in \mathcal{S}'} u_S S$ mit minimaler Delivery Time $DT = \sum_{S \in \mathcal{S}'} u_S$.

- Gegeben sei eine Teilmenge $\mathcal{S}' \subseteq \mathcal{S}$, die nicht für alle Matrizen exakte Zerlegungen ermöglicht.

  Approx-MIN-TC: Finde eine nichtnegative ganzzahlige Approximationsmatrix B, die in Segmente aus $\mathcal{S}'$ zerlegbar ist, so dass der *Total Change* $TC := \|A - B\|$ minimal ist.

- Gegeben seien eine Teilmenge $\mathcal{S}' \subseteq \mathcal{S}$, die exakte Zerlegungen aller Matrizen ermöglicht, sowie zwei nichtnegative ganzzahlige Matrizen $\underline{A}$ und $\overline{A}$. Es leiten sich zwei Probleme zur DT-Minimierung ab:

  Approx-MIN-DT: Finde eine nichtnegative ganzzahlige Approximationsmatrix B mit $\underline{a}_{ij} \leq b_{ij} \leq \overline{a}_{ij}$ für alle $(i,j) \in [m] \times [n]$, so dass die optimale Delivery Time einer Segmentierung von B in Segmente aus $\mathcal{S}'$ minimal ist.

  Approx-MIN-DT-TC: Löse **Approx-MIN-DT**, so dass DT und TC lexikographisch minimiert werden.

In der vorliegenden Arbeit werden die aufgezählten Probleme für verschiedene spezielle Mengen $\mathcal{S}'$ behandelt. Soweit nicht explizit eine andere Norm angegeben ist, wird jeweils die ℓ_1-Norm als Maß für die Abweichung verwendet.

3. Die Lamellen des Kollimators weisen kleine Überlappungen auf, um zu verhindern, dass Strahlung zwischen den Lamellen hindurch gelangt. Dies kann zu Unterdosierungseffekten im Überlappungsbereich führen. Um diese minimal zu halten, verlangt die Tongue-and-Groove-Bedingung, dass untereinander liegende Matrixeinträge maximal oft gemeinsam bestrahlt werden müssen, d.h.

$$\begin{aligned} a_{ij} \leq a_{i-1,j} \wedge s_{ij} = 1 \quad &\Longrightarrow \quad s_{i-1,j} = 1, \\ a_{ij} \geq a_{i-1,j} \wedge s_{i-1,j} = 1 \quad &\Longrightarrow \quad s_{ij} = 1 \end{aligned}$$

für $(i,j) \in [2,m] \times [n]$. Sei die Menge der zulässigen Segmente $\mathcal{S}_{TG}$. Das Problem **MIN-DT** mit $\mathcal{S}' = \mathcal{S}_{TG}$ kann mit zwei verschiedenen Minimum-Cost-Flow-Formulierungen mit Nebenbedingungen gelöst werden. Die Lösung lässt sich folglich durch Lösen eines ganzzahligen linearen Programms finden. Die Komplexität des Problems ist allerdings weiter unbekannt. Für den Fall, dass die Zielfluenz A eine binäre Matrix ist, konnte ein polynomieller Segmentierungsalgorithmus angegeben werden. Alternativ wurde das Problem auch auf ein Färbungsproblem in einem perfekten Graphen zurückgeführt.

4. Die Probleme **Approx-MIN-DT** und **Approx-MIN-DT-TC** wurden für den unrestringierten Fall $\mathcal{S}' = \mathcal{S}$ vollständig mit Hilfe kombinatorischer Algorithmen gelöst.

Die Interleaf-Collision-Bedingung verbietet ein Überlappen benachbarter linker und rechter Lamellen des Kollimators. Betrachtet man den Fall, dass die zulässigen Segmente die Interleaf-Collision-Bedingung erfüllen müssen, wurde **Approx-MIN-DT** auf eine längste-Wege-Suche in einem geeignet gewählten Graphen zurückgeführt, während **Approx-MIN-DT-TC** nur heuristisch gelöst wurde.

5. Das Problem **Approx-MIN-TC** ist für eine beliebig gewählte Menge $\mathcal{S}'$ NP-hart. Dies wird durch Reduktion vom 3SAT-6-Problem bewiesen. Hat die Matrix A nur eine Zeile oder können die Zeilen von A unabhängig zerlegt werden, so wird eine Minimum-Cost-Flow-Formulierung angegeben, welche die polynomielle Lösbarkeit in diesem Fall beweist.

6. Im klinischen Alltag sollen die Segmente aus dosimetrischen Gründen gewissen Mindestgrößenanforderungen genügen. Es wurde ein umfangreiches praxisnahes Modell erlaubter Segmente erstellt und eine Heuristik für **Approx-MIN-TC** (mit einer zusätzlichen Nebenbedingung) in diesem Fall entwickelt. Die Ergebnisse waren sehr überzeugend. Es konnten trotz signifikanter Werte des Total Change in der Segmentierung Behandlungspläne mit vergleichbarer Qualität zur exakten Segmentierung, allerdings deutlich weniger Segmenten (und geringerer Delivery Time) erstellt werden. Damit kann die Behandlung wesentlich effizienter gestaltet werden.

7. Für das Problem **Approx-MIN-TC** wird ein allgemeiner Lösungsalgorithmus mit Hilfe von Column-Generation-Methoden jeweils for die ℓ_1-Norm und die ℓ_2-Norm entwickelt.

8. Schließlich wird den diskreten Segmentierungsproblemen ein kontinuierlicher Segmentierungsansatz entgegengestellt, der in der Lage ist, die physikalischen Eigenschaften der Strahlung besser zu modellieren. Die Modellannahmen der diskreten Verfahren, dass Fluenzverteilungen ganzzahlige Treppenfunktionen darstellen, sind aufgrund von Halbschatteneffekten der Strahlung in der Praxis nicht erfüllt. Es erfolgt eine realitätsnahe Modellierung der Fluenzverteilungen und der Segmente als kontinuierliche Funktionen $f : [0, m] \times [0, n] \to \mathbb{R}_+$ sowie $S : [0, m] \times [0, n] \to [0, 1]$. Dazu werden verschiedene Decline-Funktionen eingeführt, die den Halbschatten der Strahlung abbilden. Zudem wird ein additives Fluenzmodell beschrieben, welches es ermöglicht, mit einer kleinen Segmentanzahl $|\mathcal{S}'|$ zu arbeiten. Die Abweichung zwischen realisierter Fluenz und Zielfluenz wird mit der L_2-Norm gemessen. Die kontinuierliche Form des Problems **Approx-MIN-TC** ist dann

$$\textbf{Approx-MIN-TC-Continuous:} \qquad \Big\| f - \sum_{S \in \mathcal{S}'} u_S S \Big\|_2 \to \min$$
$$u_S \geq 0 \quad \forall S \in \mathcal{S}'.$$

Es wird gezeigt, dass dies äquivalent zur Lösung eines quadratischen Optimierungsproblems mit Nichtnegativitätsbedingungen der Form

$$h(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T D \boldsymbol{u} + \boldsymbol{c}^T \boldsymbol{u} \to \min \quad \text{mit} \quad \boldsymbol{u} \geq \boldsymbol{0}$$

ist.

9. Integriert man zusätzlich in das kontinuierliche Fluenzmodell, dass ein Teil der Strahlung durch die Lamellen hindurch gelangt (Leckstrahlung), muss die Additivität des zuvor genannten Modells aufgegeben werden. Dadurch erhöht sich $|\mathcal{S}'|$ dramatisch. Analog zum diskreten Fall werden wieder Column-Generation-Methoden zur Problemlösung entwickelt.