An Integer Programming Approach to the Multileaf Collimator Problem

Diploma Thesis

by

Frank Lenzen

Supervisor

Prof. Dr. Horst W. Hamacher

University of Kaiserslautern, Dept. of Mathematics
P.O. Box 3049, D-67653 Kaiserslautern
Germany

Kaiserslautern, June 2000

Preface

First of all, I would like to express my gratitude to Prof. Dr. Horst W. Hamacher, who gave me the possibility to start working in this fascinating field as a research assistant at the Institut für Techno und Wirtschaftsmathematik in 1998. Furthermore he convinced me of doing more research under this topic in the framework of my final thesis. Not only did he always believe in a good outcome of my work even during hard times where drawbacks had to be faced, but supported me in any possible way. In this context I would like to thank all members of his group at the Department of Mathematics at the University of Kaiserslautern and at the Institut für Techno und Wirtschaftsmathematik, who created a warm and friendly atmosphere.

Last, but not least, I would like to thank my friends and parents for their great support.

Contents

Chapter 1

Introduction

1.1 Cancer, Radiation Therapy and Optimization

Every year more than 340,000 people in Germany are diagnosed with cancer - and more than 210,000 of them die [1]. More than 70,000 of all diagnosed cancer patients suffer from an uncontrolled growth of the primary tumor [Bortfeld (1995)]. In [Peregrine] it is reported that in the United States of America about 50 % of all cancer patients die due to a disease at and around the primary tumor site. A situation which is far from being satisfactory concerning the success in cancer treatment.

Within the past years mathematical methods have been developed to support physicians in their fight against cancer. Radiation therapy is a particular field where methods of operations research are currently being developed. Radiation therapy is a commonly used means to fight cancer when the tumor can be localized and metastases have not yet started to form. Its purpose is to supply enough energy to the tumor such that either all clonogenic cells are destroyed or the tumor can at least be controlled in its growth. The fact that the tumor surrounding organs – the *organs at risk* – are in general very sensitive to radiation is a pressing problem in the planning process. In order to completely understand the planning stage of radiation therapy the treatment setup shall be explained in more detail.

The radiation is supplied by a medical linear accelerator, which can be rotated around the patient who is positioned and fixed in a stereotactical mask on a couch (cf. Figure 1.1).

Figure 1.2 represents a two dimensionsal slice of the human body. The tumor will from now on be called the *target volume* - it will be the target which shall be harmed as much as possible. The goal of radiation therapy planning is to create a plan which describes how to setup the treatment such that the target volume receives the prescribed dose without harming the surrounding organs. The planners have to identify the number of beams, their directions and the isocentre – i.e. a single point – in which all beam directions meet. Given the number n of beams, finding the beam directions and the isocenter can mathematically be formulated as an $n+1$ facility location problem: n locations for the beams and 1 for the isocenter. In addition, the plan has to include a good intensity function for each of the beam directions. It is of major importance to know that doses delivered at different times to a single point in the human body

[1] cf. [Atlas of cancer mortality]

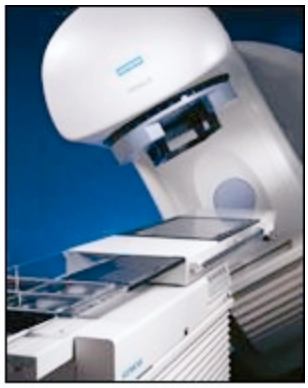


Figure 1.1: *A medical linear accelarerator with a beam head and a treatment couch*

add linearly to the overall dose delivered to that point.

But the process of irradiating the target volume is restricted by the sensitivity to radiation of the surrounding organs. Delivering too much radiation to the surrounding organs includes a high risk of harming these organs' health and functions. This is in general an unsolvable conflict and the main reason for the multiobjective approach in the planning stage [Hamacher & Küfer (1999)].

The unsuccessful application of radiation therapy in some cases is due to the inability to deliver the required irradiation to the target volume without violating the constraints which are set by the sensitivity to radiation of the organs at risk.

After a good plan has been found the task is to treat the patient according to the given plan. Due to technological restrictions this does actually give rise to other problems. The major point is, that a linear accelerator can only generate a cone-shaped field of radiation with a homogeneous intensity. Today, there are two important methods which enable us to deliver the required intensity function. These two approaches will be described in the next section.

Before starting the analysis of the two methods in the next section, a brief introduction of the layout of the beam's head will be given. The beam's head can also be seen in Figure 1.1 as the far end of the gantry which focuses the radiation on the required target area. In Figure 1.3 a cut through the beam's head is displayed. Here we can see how the radiation is bent and focused and then a cone shaped field is created which is directed towards the tumor in the patient. The *beam's eye view* is the view a person has when watching from the radiation focus, that is the cone's peak, towards the patient.

The output of the treatment planning process is a matrix $\mathcal{I} \in \mathbb{Z}^{m \times n}$ for each beam direction, where $m, n \in \mathbb{Z}$ are usually not less than 10 and not larger than 20, which describes the discrete intensity function on the beam's head.

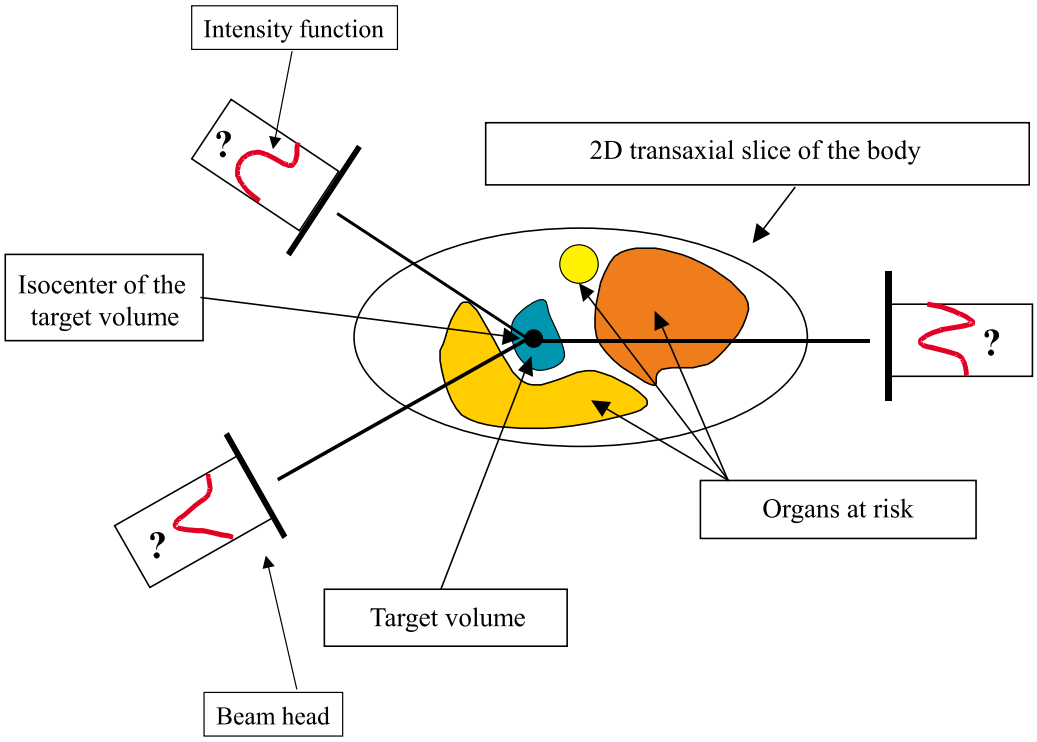


Figure 1.2: *The treatment planning. Here $n = 3$ and the isocenter is located in the center of mass of the target volume.*

1.2 Generating Intensity Modulated Fields

1.2.1 Metal Compensators

The idea of using metal compensators to modulate the intensity is very simple. Making use of electron- or x-ray-absorbing material (e.g. lead), respectively, of varying thickness, it is then possible to modulate the intensity by creating a three-dimensional object (compensator) of this material - the shape of which is related to the given intensity matrix - by mounting this object to the beam-head. Locations where the absorbing material is very thin results in quite a high intensity behind this part of the compensator. On the other hand, the intensity of the radiation behind those locations where the compensator's thickness is large, is considerably low. The major disadvantage is that each compensator is uniquely manufactured for a single gantry angle and for a single patient - clearly because the geometry of the target volume and the organs at risk and of course the dose requirement are depending significantly on the patient.

The major advantage of this approach in modulating intensities is the fact that one can test these compensators in advance (e.g. in a laboratory), i.e. before the treatment actually starts and measure the delivered dose in order to avoid failures in the delivery and to detect manufacturing errors. But a statistical survey showed that small errors are often neglected when they are detected in the hospital and when the patient is already preparing for his treatment [Mohan (1995)]. Of course, the production of a metal compensator is time consuming and labour intensive which results in the fact that, due to costs, the planning process is limited to only a few number of gantry angles. This can affect the quality of the treatment plan enormously. In

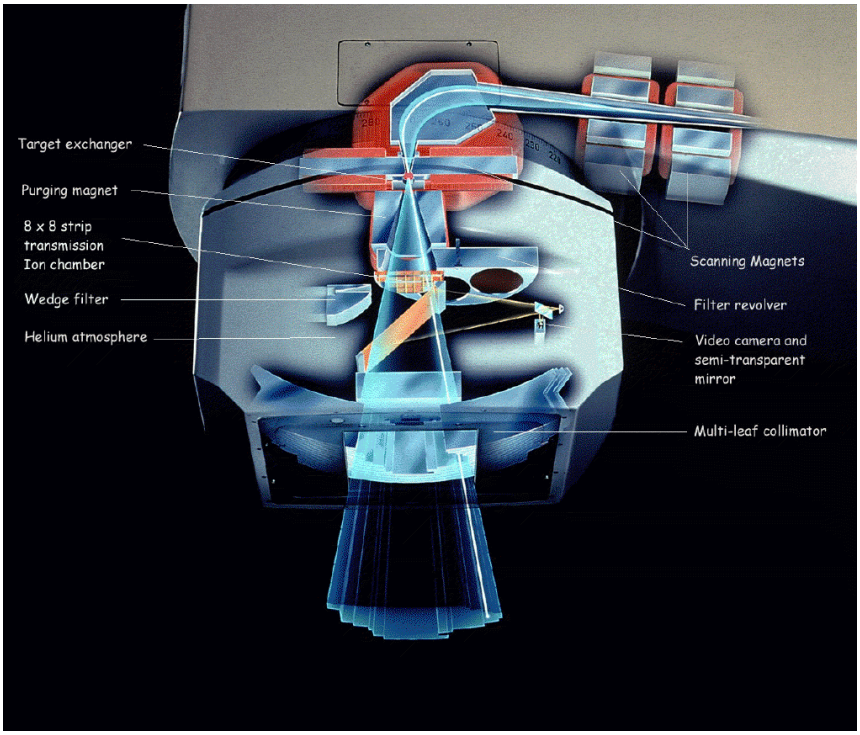


Figure 1.3: *Cut through a beam's head*

addition, there is a significant risk of patient injury due to a falling block when a technician is about to change the compensator after the irradation has been finished at a certain gantry angle. Additionally, this technique can cause prolonged treatment times when a complex plan has to be delivered, since the technicians have to enter the treatment room whenever the compensator needs to be changed [Galvin et al.(1993)].

1.3 Multileaf Collimators

The usage of a multileaf collimator (MLC) is very different from the compensator-approach. Of course, absorbing material is used as well, but the way how this is done is totally different.

Referring to Figure 1.5 one uses high metal pieces, the height of which (usually around 5 to 7 cm) totally blocks any radiation. These pieces are called leaves as their thickness d_{leaf} is very low, usually only 5 to 10 mm. Two of those leaves are placed opposite to each other. Each of them is connected to a linear motor by a metal band and can move in the direction towards the other leaf or away from it. Such two leaves are called a *channel* or a *row*. Placing several leaf pairs as described above adjacent to each other as in Figure 1.6 we can shape a two-dimensional area which then - when irradiating this area for a certain time - receives a certain dose. The blocked area does, theoretically, not receive any radiation at all.

There are two different methods of the MLC's usage which will now be explained.

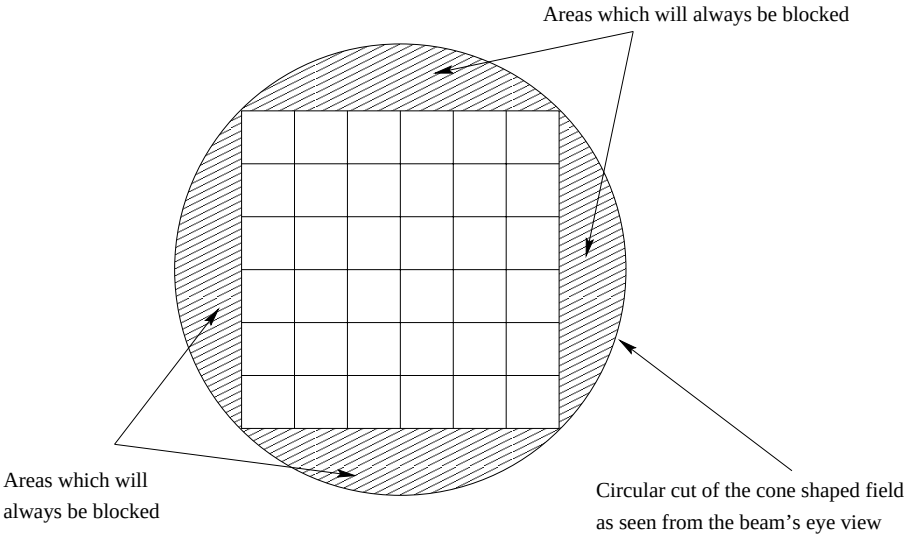


Figure 1.4: *The discretized area seen from the beam's eye through the MLC when all leaves are totally retracted. In this figure the leaves are retracted to the left and right side, respectively.*

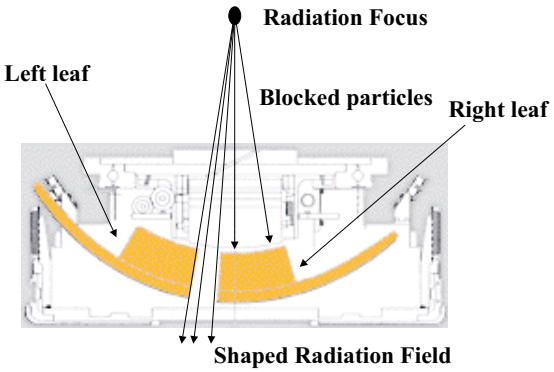


Figure 1.5: *A leaf pair seen by a cut through the MLC*

1.3.1 The Dynamic Mode

The dynamic mode can be described as follows: The leaf-pairs are positioned at an initial position and then the beam is turned on. Then the leaves move with a calculated, not necessarily constant, speed while the beam remains switched on in order to create the desired intensity profile. An example helps a lot to understand this technique. Suppose we have a single-channel MLC, i.e. a single leaf pair. Suppose the desired intensity profile is $(0, 1, 2, 1, 0, 0)$. Position the left leaf initially to the left of column 2 and the right leaf to the left of column 4. An additional parameter is the doserate, which is here assumed to be equal to $1\frac{Gy}{sec}$. Moving the leaves with a constant speed of $0.5\frac{columns}{sec}$ to the right for a total time of $2sec$ and finally switching off the beam, then the mean dose in each column equals the desired intensity profile.

1.3.2 The Static Mode

The static mode is of course different from the dynamic mode, but it does as well exploit the fact that doses delivered at different times to the same destination sum up linearly. The idea

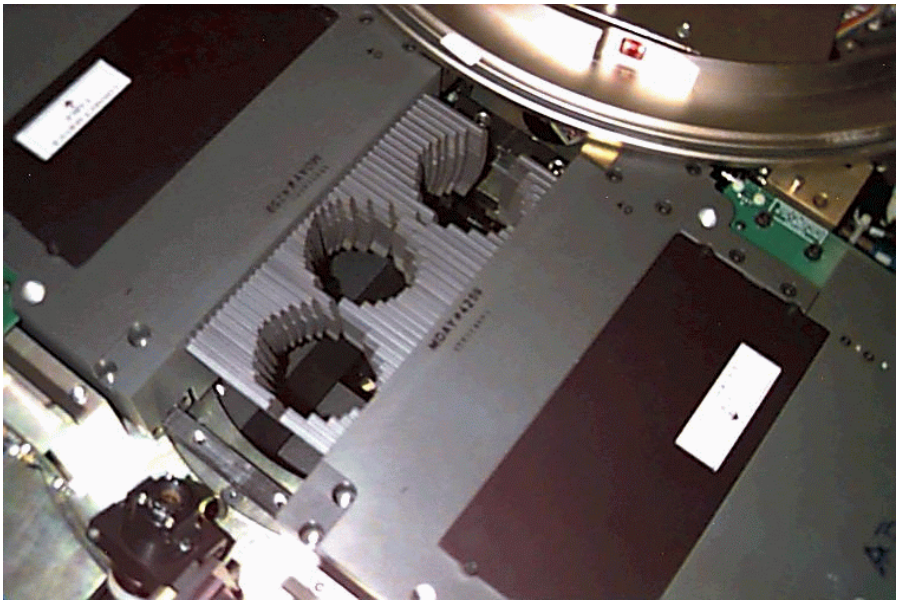


Figure 1.6: *The adjacent leaf pairs of an MLC seen in an openend beam head*

is to switch off the beam while the leaf pairs are being moved to their desired position. Then, keeping the leaf pairs at this position the beam is switched on for a certain time in order to irradiate the area, which is not blocked by the MLC leaf pairs. This procedure is repeated until the required intensity profile has been delivered.

Thus one is looking for a feasible decomposition of $\mathcal{I}$ such that

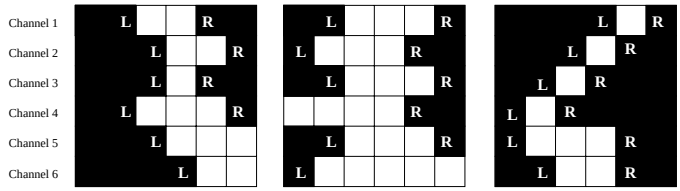
$$\mathcal{I} = \sum_k M^k, \; M^k \in \; \{0,1\}^{m \times n}.$$

A 1-entry corresponds to an uncovered area, where as if $M^k_{ij} = 0$ means, that either the right or the left leaf of channel i covers the j-th column. The underlying constraints to create a matrix M^k will be explained later. A first example shall help in understanding the relation between matrices and the MLC leaf sequencing.

Let

$$\mathcal{I} = \begin{pmatrix} 0 & 0 & 2 & 2 & 2 & 0 \\ 0 & 1 & 1 & 3 & 1 & 0 \\ 0 & 0 & 2 & 2 & 1 & 0 \\ 1 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 2 & 1 \\ 0 & 1 & 2 & 2 & 2 & 2 \end{pmatrix}$$

then we can get the following MLC leaf sequence in order to shape this intensity profile.



The black areas are covered either by a left (L) or a right (R) leaf as indicated in the figure, respectively. This can be written as the sum of $M^1 + M^2 + M^3$, as follows:

$$\mathcal{I} = M^1 + M^2 + M^3$$

$$\mathcal{I} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

The following work will concentrate on the static mode decomposition technique - not only due to the fact that controlling and veryfying the delivery of irradiation in the dynamic mode is indeed a very hard problem [Bortfeld et al.(1994)], but we are more interested in a combinatorial optimization problem. The leaf-ends will always be positioned such that their location corresponds to the vertical grid line between two adjacent cells of the matrix in the given channel. From now on we will always say that a leaf is positioned to the left of a cell if the leaf-end corresponds to a location which is touching the left vertical gridline of this cell. This will be very important later on.

Chapter 2

Restrictions and The Objective

2.1 Restrictions

The decomposition of the intensity matrix $\mathcal{I}$ into shape matrices M^k is in fact a combinatorial optimization problem, with certain restrictions and different objectives. First of all, a short overview of the most common restrictions is given. It should be taken care of the fact that the restrictions vary among the manufacturers of the MLCs and thus results of various models are often incomparable. Nevertheless it can be observed that a common set of restrictions is being discussed in all major publications.

- *Interleaf motion constraints*
  Interleaf motion constraints prohibit the movement of a left leaf further to the right than any of its opposing right leaves in the adjacent channels. This constraint is due to technical requirements of the leaves. The leaves are designed in such a way that, when positioned adjacent to each other, as few radiation as possible "leaks" in between. Thus two adjacent leaves are touching each other. Suppose a left or right leaf has moved further to the right or left, respectively, than any of its opposing leaves in an adjacent channel. There shall be no leakage in between two adjacent leaves. Thus when the left or right leaf, respectively, is about to move beyond the end of the opposing leaf than the leaf-ends either crash or there is a significant leakage in between the two leaves in the process of irradiation [Mohan (1995)], [Webb (1998b)], [Xia & Verhey (1998)], [Siochi (1999)].

- *Abutting leaf-ends*
  This constraint is to forbid the closure of a channel during the phase of irradiation whilst others remain open for irradiation. The reason why this constraint is modelled sometimes is due to a high leakage through the abutting leaf-ends [Webb (1998b)] but this is depending on the manufacturers of the MLC. Since Webb was the only one with such a constraint it has been decided to drop this one.

- *Tongue-and-groove effects*
  Tongue-and-groove effects occur due to the special design of the flanks of each leaf. In order to reduce the leakage of radiation in between two adjacent leaves the flanks are covered by a tongue-and-groove design [Mohan (1995)], [Webb (1998b)], [Xia & Verhey (1998)], [Siochi (1999)]. In return, this causes an underdosage along the grid line next to which no leaf is positioned. The effect can be reduced when irradiating adjacent cells simultaneously. This is of course only possible to a certain extent as the cells might have different intensity

requirements stated in the intensity matrix $\mathcal{I}$. This does actually give rise to the following: The tongue-and-groove effect can be reduced in a given decomposition of $\mathcal{I}$, if there exists matrices M^k and M^l such that $M^k_{ij} = 1$, $M^k_{i+1,j} = 0$ where $M^l_{ij} = 0$ and $M^l_{i+1,j} = 1$. The two constraints mentioned previously are hard constraints which had to be fulfilled when being modelled. It is an open question whether the effect should be eliminated or just reduced to a certain level [Webb (1998b)].

2.2 The Objective

We have still not discussed the objective of the decomposition, which is more than just finding a feasible decomposition. The problem with the decompositions is at hand: having large and complex intensity matrices $\mathcal{I}$ it is not easy to find a *good* decomposition, i.e. a decomposition which minimizes the overall delivery time. The overall delivery time is *not* just proportional to the number of segments used in the decomposition which is basically the beam-on time. We have to consider that in between two non-identical segments the beam has to be switched off and the configuration of the MLC has to be changed. It should be taken care of, that between two identical segments there is in general no additional setup time. Here, the irradiation continues without a break. Extending our model of decomposing $\mathcal{I}$ into matrices M^k we are most keen on finding a decomposition of the form

$$\mathcal{I} = \sum_{k=1}^{N_{Seg}} \alpha_k M^k, \text{ where } \alpha_k \in \mathbb{Z}, \ \forall k = 1, \ldots, N_{Seg}$$

where N_{Seg} is the unknown number of segments, which minimizes

$$\sum_{k=1}^{N_{Seg}} \alpha_k * C_{D_R} + \sum_{k=1}^{N_{Seg}-1} T^k_{Setup},$$

and $C_{D_R} \geq 0$ is a constant factor, depending on the specifications of the linear accelerator. α_k is called the relative beam-on time coefficient, as $\alpha_k * C_{D_R}$ determines the beam-on time for shape matrix M_k, i.e. the time for how long the beam is switched on using the MLC setup according to shape matrix M_k. [Siochi (1999)] has observed that the setup-time between two non-identical segments T^k_{Setup} is not constant but is depending on the shape of the preceding and the following setup defining segments. Furthermore it can be seen that for some MLC the setup-time between two non-identical segments is defined by

$$T^k_{Setup} := \max \left\{ OH_{VR}, \max_{i \in \{1,\ldots,m\}} \left\{ \max \left\{ T^L_{Travel}(i,k), T^R_{Travel}(i,k) \right\} \right\} \right\}$$

where

- OH_{VR} is the verify and record overhead which is needed to verify the positions of the leaves and to record them via portal imaging techniques. Usually OH_{VR} is between 4 and 18 *sec.*
- $T^L_{Travel}(i,k)$ is the time which the left leaf in channel i needs in order to move from the old position in segment k to the new one in the following segment $k+1$.
- $T^R_{Travel}(i,k)$ is the time which the right leaf in channel i needs in order to move from the old position in segment k to the new one in the following segment $k+1$.

The derivation of T_{Setup} has been extensively studied in [Siochi (1999)].

Chapter 3

Mathematical Modelling

3.1 A First Mixed Integer Programming Model

A first mixed integer programming model will help us in understanding the possibilities and drawbacks of a mathematical model which is basically "translated" into equations derived from reality.

In the following the variables which will be needed for the model are defined. Let

$$y_{i,j,t} := \begin{cases} 0 & \text{if the } t\text{-th shape matrix covers the } j\text{-th cell in channel } i \\ 1 & \text{otherwise} \end{cases}$$

$$L_{i,j,t} := \begin{cases} 1 & \text{if the left leaf in channel } i \text{ of shape matrix } t \text{ is positioned to the left of cell } j \\ 0 & \text{otherwise} \end{cases}$$

$$R_{i,j,t} := \begin{cases} 1 & \text{if the right leaf in channel } i \text{ of shape matrix } t \text{ is positioned to the left of cell } j \\ 0 & \text{otherwise} \end{cases}$$

It should be stressed that $y_{i,j,t} = 1$ corresponds to irradiating the j-th cell in the i-th channel using to the t-th shape matrix. The restrictions which have been explained in the previous section can now be modelled.
We start with intuitively necessary constraints which will help to create a physical solution of the model. A physical shape matrix is a matrix which can in general be shaped by the MLC. Each constraint shall be explained explicitely in order to ensure a full understanding of the model which will be needed later on. The reader should be convinced that an $m \times n$ intensity matrix $\mathcal{I}$ has $n+1$ possible L- and R- leaf positions in each channel, namely to the left and right of each column of the matrix. Furthermore, we define $\mathcal{T} := \{1, \ldots, N_{Seg}\}$.

- First of all, there must be exactly one L- and R-leaf position defined in every channel, which gives rise to the following two constraints:

$$\sum_{j=1}^{n+1} L_{i,j,t} = 1 \quad \forall\, t \in \mathcal{T},\ \forall\, i \in \{1, \ldots, m\} \tag{3.1.1}$$

$$\sum_{j=1}^{n+1} R_{i,j,t} = 1 \quad \forall\, t \in \mathcal{T},\ \forall\, i \in \{1, \ldots, m\} \tag{3.1.2}$$

- In addition, we have to ensure, that the L-leaf of a channel i is not positioned further to the right then the R-leaf in the same channel:

$$\sum_{j=1}^{n+1} j * L_{i,j,t} \leq \sum_{j=1}^{n+1} j * R_{i,j,t} \quad \forall t \in \mathcal{T},\ \forall i \in \{1,\dots,m\} \tag{3.1.3}$$

- In order to forbid interleaf motion, the following constraints are added:

$$\sum_{j=1}^{n+1} j * L_{i,j,t} \leq \sum_{j=1}^{n+1} j * R_{i-1,j,t} \quad \forall\ t \in \mathcal{T},\ i \in \{2,\dots,m\} \tag{3.1.4}$$

$$\sum_{j=1}^{n+1} j * L_{i,j,t} \leq \sum_{j=1}^{n+1} j * R_{i+1,j,t} \quad \forall\ t \in \mathcal{T},\ i \in \{1,\dots,m-1\} \tag{3.1.5}$$

- Last but not least, in order to ensure the transferability to reality, it is of major importance that:

$$L_{i,j,t},\ R_{i,j,t} \in \{0,1\} \quad \forall\ i \in \{1,\dots,m\},\ j \in \{1,\dots,n+1\},\ t \in \mathcal{T} \tag{3.1.6}$$

So far, the leaves have been placed such that the shape matrices are physically deliverable. It is now necessary to link the position of the L- and R-leaves with the irradiated area, which is defined by the variables $y_{i,j,t}$.

- When the L- and R-leaf of channel i in segment t do not share the same position j, and the L-leaf is positioned to the left of cell j, then cell j will be irradiated.

$$y_{i,j,t} \geq L_{i,j,t} - R_{i,j,t} \quad \forall\ i \in \{1,\dots,m\},\ j \in \{1,\dots,n+1\},\ t \in \mathcal{T} \tag{3.1.7}$$

- Consequently, a similar statement holds, when the R-leaf is positioned to the left of cell $j+1$ and the L-leaf does not occupy the same position, then cell j will be irradiated:

$$y_{i,j,t} \geq R_{i,j+1,t} - L_{i,j+1,t} \quad \forall\ i \in \{1,\dots,m\},\ j \in \{1,\dots,n\},\ t \in \mathcal{T} \tag{3.1.8}$$

- On the other hand we do need constraints that force L- and R-leaf positions, respectively, whenever there is a so called *step* in a row of a shape matrix. A step in a row at position j is a situation when cell j is irradiated and cell $j+1$ is not - or vice versa. This will be a frequent appearance in the shape matrices, and one should convince oneself that there is either no step or two steps in a row.

$$y_{i,j,t} - y_{i,j-1,t} \leq L_{i,j,t} \quad \forall\ i \in \{1,\dots,m\},\ j \in \{2,\dots,n+1\},\ t \in \mathcal{T} \tag{3.1.9}$$

$$y_{i,j-1,t} - y_{i,j,t} \leq R_{i,j,t} \quad \forall\ i \in \{1,\dots,m\},\ j \in \{2,\dots,n+1\},\ t \in \mathcal{T} \tag{3.1.10}$$

- For completion, two additional constraints are needed to define the leaf setting at each side of the channel. Note that the two constraints (3.1.9) and (3.1.10) state conditions only between cells. The left and the right hand side are excluded as there is no cell to the left and right, respectively. Here, we force the L-leaf to be positioned to the left of cell 1, if cell 1 is irradiated. A similar statement holds for the last cell: when cell n is irradiated then the R-leaf has to be placed to the right of cell n, i.e. to the left of the non existing cell $n+1$:

$$y_{i,1,t} \leq L_{i,1,t} \quad \forall\ i \in \{1,\dots,m\},\ t \in \mathcal{T} \tag{3.1.11}$$

$$y_{i,n,t} \leq R_{i,n+1,t} \quad \forall\ i \in \{1,\dots,m\},\ t \in \mathcal{T} \tag{3.1.12}$$

- Of course, the "binary" constraint for the y-variables is needed:

$$y_{i,j,t} \in \{0,1\} \quad \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\},\ t \in \mathcal{T} \tag{3.1.13}$$

Finally, an additional constraint ensures the delivery of the required intensity profile, that is

$$\sum_{t\in\mathcal{T}} y_{i,j,t} = \mathcal{I}_{i,j} \quad \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\} \tag{3.1.14}$$

All necessary constraints in order to decompose the required intensity profile into valid shape matrices have been modelled, but the objective as discussed earlier is still missing. Therefore it is necessary to define an additional set of variables to model the objective function:

$$m_t := \begin{cases} 1 & \text{if at least one leaf pair is not closed in the } t\text{-th shape matrix} \\ 0 & \text{otherwise} \end{cases}$$

$$reduce_t := \begin{cases} 1 & \text{if segment } t \text{ and segment } t+1 \text{ are not identical in shape} \\ 0 & \text{otherwise} \end{cases}$$

$$setup_t := \max\left\{OH_{VR}, \max_{i\in\{1,\dots,m\}}\left\{\max\left\{T^L_{Travel}(i,t), T^R_{Travel}(i,t)\right\}\right\}\right\}$$

Comparing the definition $setup_t$ and T^k_{Setup} in the previous chapter identifies the formula for $setup_t$ as the maximal value of overhead and leaf-travel time. In order to identify the variables as in their definition the following constraints have to be added to the model:

- First of all, it is easy to see, that the beam is in use in shape matrix t, whenever at least a single channel is not closed. Thus, the definition of m_t can be represented by the following two constraints

$$m_t \geq y_{i,j,t} \quad \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\},\ t \in \mathcal{T} \tag{3.1.15}$$

$$m_t \in \{0,1\} \quad \forall\, t \in \mathcal{T} \tag{3.1.16}$$

- Similarly $reduce_t$ can be defined by a slight change in the definiton of m_t, namely

$$reduce_t \geq |y_{i,j,t} - y_{i,j,t-1}| \quad \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\},\ t \in \mathcal{T}\setminus\{1\} \tag{3.1.17}$$

$$reduce_t \in \{0,1\} \tag{3.1.18}$$

  One can observe that $reduce_t = 1$ if there is a change between shape matrix t and $t+1$. It will be more convenient to handle only linear constraint and thus the nonlinear constraint (3.1.17) is replaced by the following linear constraints:

$$reduce_t \geq y_{i,j,t} - y_{i,j,t-1} \quad \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\},\ t \in \mathcal{T}\setminus\{1\} \tag{3.1.19}$$

$$reduce_t \geq y_{i,j,t-1} - y_{i,j,t} \quad \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\},\ t \in \mathcal{T}\setminus\{1\} \tag{3.1.20}$$

In this model we will use a constant setup time for simplicity, that is $setup_t := OH_{VR}\ \forall\, t \in \mathcal{T}$.

The first model is then:

$$(MIP)\qquad \begin{array}{ll} \min & \sum_{t\in\mathcal{T}} m_t * C_{D_R} + OH_{VR} * reduce_t \\ \text{s.t.} & (3.1.1), (3.1.2), (3.1.3), (3.1.4), (3.1.5), (3.1.6) \\ & (3.1.7), (3.1.8) \\ & (3.1.9), (3.1.10), (3.1.11), (3.1.12) \\ & (3.1.14) \\ & (3.1.13) \\ & (3.1.15), (3.1.16), (3.1.19), (3.1.20), (3.1.18) \end{array}$$

Additionally, this model is very handy in terms of modelling additional constraints as, e.g. the tongue-and-groove constraints to completely remove this undesired effect. Tongue-and-groove effects occur when adjacent cells in different channels are irradiated alternately. Speaking in terms of the previously defined variables, the effect occurs, when $y_{i,j,t} = 1, y_{i-1,j,t} = 0$ and $y_{i,j,t^*} = 0, y_{i-1,j,t^*} = 1$, where t and t^* are the indices of two different shape matrices. A careful analysis yields:

Theorem: The tongue and groove effect occurs, if and only if one of the following constraints is violated for any two shape matrices $t, t^* \in \mathcal{T}, t \leq t^* - 1$:

$$\begin{aligned} (y_{i,j,t} - y_{i-1,j,t}) - (y_{i,j,t^*} - y_{i-1,j,t^*}) &\geq -1 \quad \forall\, i \in \{2,\ldots,m\},\ j \in \{1,\ldots,n\} \\ (y_{i,j,t} - y_{i-1,j,t}) - (y_{i,j,t^*} - y_{i-1,j,t^*}) &\leq 1 \quad \forall\, i \in \{2,\ldots,m\},\ j \in \{1,\ldots,n\} \end{aligned}$$

Proof: The only possible configurations are given in the table below $[a, b, c \in \{0, 1\}]$:

$y_{i,j,t}$	$y_{i-1,j,t}$	y_{i,j,t^*}	y_{i-1,j,t^*}	$(y_{i,j,t} - y_{i-1,j,t}) - (y_{i,j,t^*} - y_{i-1,j,t^*})$	Tongue-and-groove
0	1	1	0	-2	yes
1	0	0	1	2	yes
a	a	b	c $(b \neq c)$	$-1 \leq c - b \leq 1$	no
b	c $(c \neq b)$	a	a	$-1 \leq b - c \leq 1$	no
a	b	a	b	0	no

q.e.d.

A very severe drawback using this model is based on the set of segments $\mathcal{T} = \{1,\ldots,N_{Seg}\}$. As the number of segments N_{Seg} is not known beforehand, one needs to compute an upper bound. Computing a good upper bound is easy for small and easily realizable plans, but for complex plans it is a very hard problem, actually it is as hard as the original problem. I.e. a lot of segments will not be used when applying this model. So far, one conclusion is, that far too many variables, and thus constraints, have been included in our model. This gives rise to problems in the CPU time, which will grow very fast even for small instances of the problem.

3.2 A Column Generation Approach

This is a completely new approach to the problem and the initial idea was first brought up by a participant of the third ALIO-EURO Workshop on Applied Combinatorial Optimization in Erice, Italy, 1999. The idea is as follows:

Defining $\mathcal{U} \in \{0,1\}^{m*n\times r}$ as the matrix the columns of which correspond to single shape matrices, each written row-wise as a column. r is the total number of columns which are feasible for the MLC. A small example illustrates this easily: The shape matrix

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

corresponds to a column k of $\mathcal{U}$, which is defined as follows:

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

In a similar way we transform the intensity matrix $\mathcal{I}$ into a column and we gain $\underline{\mathcal{I}}$. One can easily conclude that r increases very fast, and for $m = n = 10$ we can compute r to be of the order 10^{19}. But for a first analysis we simply neglect this and concentrate on the theoretical problem description. Now, the problem can be stated in a very compact way, that is

$$(P_{Master}) \quad \begin{array}{ll} \min & \underline{1}^T\alpha \\ \text{s.t.} & \mathcal{U}\alpha = \underline{\mathcal{I}} \\ & \alpha_i \in I\!R \ , \alpha_i \geq 0 \quad \forall\ i \in \{1,\ldots,r\} \end{array}$$

where $\underline{1}$ is the vector of ones. As can be seen easily, the objective function does no longer include setup times in between two non-identical segments. It would theoretically be possible to model it here, but this revised formulation will be given later. The objective function which is given above basically reduces the beam-on time. It was claimed in [Siochi (1999)] that minimizing the sum of the relative beam-on time coefficients α_i is a good measure for minmizing the total delivery time. We will comment on this later on. A common approach to solve these systems is the column generation technique. The major advantage using this technique together with the revised Simplex method is that only those columns which correspond to the basis variables have to be stored explicitely. Running the revised simplex algorithm, one needs to find a variable with negative reduced cost in order that this variable can either enter the basis or an unbounded polyhedron is detected. It should be noticed that from our application our polyhedron cannot be unbounded. Enumerating and computing all reduced costs would be tedious if not impossible

when r is exponentially large. Thus, the column-generation subproblem is asking for the column q with minimal reduced costs:

$$\left(P_{CG}^{general}\right) \quad \begin{array}{ll} \min & z = 1 - \underline{1}_B^T \mathcal{U}_B^{-1} \mathcal{U}_{.q} \\ \text{s.t.} & q \in N \end{array}$$

where N is the set of nonbasic variables, $\underline{1_B}$ is the vector of ones of dimension $m*n$ corresponding to the costs of the basis variables, $\mathcal{U}_B$ is the submatrix of $\mathcal{U}$ correponding to the current basis matrix of the master problem defined above, and $\mathcal{U}_{.q}$ is the column of $\mathcal{U}$ which corresponds to the q-th variable. For a detailed description one can refer to [Lasdon].

It can easily be seen, that the integer property of α is not needed and thus can be relaxed. [Siochi (1999)] has observed that minimizing the sum over the relative beam-on time coefficitents ($\sum_k \alpha_k$) is a good measure of minimizing the number of segments used. The sum of the relative beam-on time coefficients is identical to our objective function, this is the reason for the choice of α as the name for the variables in the master problem.

An open question until now, is how the constraints of the column generation subproblem do look like. By definition, the column generation problem outputs a valid shape matrix of minimal reduced costs. Thus, we can make use of the constraints which we have derived in the previous section to describe the polyhedron defining the set of all valid shape matrices. It is a big advantage that the third dimension of all variables in defining the shape matrices, time, is not needed in the subproblem, as every time only one shape matrix needs to be computed. The number of variables and constraints decreases enormously. For completeness, the subproblem

and all constraints are given explicitly below.

$$
(P_{CG})\qquad
\begin{array}{lllll}
\min & 1 - \underline{1}_B^T \mathcal{U}_B^{-1} y & & & \\
\text{s.t.} & \sum_{j=1}^{n+1} L_{i,j} & = & 1 & \forall\, i \in \{1,\dots,m\} \\
& \sum_{j=1}^{n+1} R_{i,j} & = & 1 & \forall\, i \in \{1,\dots,m\} \\
& \sum_{j=1}^{n+1} j * L_{i,j} & \leq & \sum_{j=1}^{n+1} j * R_{i,j} & \forall i \in \{1,\dots,m\} \\
& \sum_{j=1}^{n+1} j * L_{i,j} & \leq & \sum_{j=1}^{n+1} j * R_{i-1,j} & \forall\, i \in \{2,\dots,m\} \\
& \sum_{j=1}^{n+1} j * L_{i,j} & \leq & \sum_{j=1}^{n+1} j * R_{i+1,j} & \forall\, i \in \{1,\dots,m-1\} \\
& y_{i,j} & \geq & L_{i,j} - R_{i,j} & \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n+1\} \\
& y_{i,j} & \geq & R_{i,j+1} - L_{i,j+1} & \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\} \\
& y_{i,j} - y_{i,j-1} & \leq & L_{i,j} & \forall\, i \in \{1,\dots,m\},\ j \in \{2,\dots,n+1\} \\
& y_{i,j-1} - y_{i,j} & \leq & R_{i,j} & \forall\, i \in \{1,\dots,m\},\ j \in \{2,\dots,n+1\} \\
& y_{i,1} & \leq & L_{i,1} & \forall\, i \in \{1,\dots,m\} \\
& y_{i,n} & \leq & R_{i,n+1} & \forall\, i \in \{1,\dots,m\} \\
& y_{i,j} & \in & \{0,1\} & \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\} \\
& L_{i,j},\ R_{i,j} & \in & \{0,1\} & \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n+1\}
\end{array}
$$

It should be taken care of the fact, that the subproblem shall output a shape matrix, written as a vector, in order to meet the requirements of the master problem, therefore after having solved the subproblem above we can easily define a new variable $COL \in \{0,1\}^{m*n}$ as follows:

$$COL_{(i-1)*n+j} = y_{i,j} \quad \forall\, i \in \{1,\dots,m\},\ j \in \{1,\dots,n\}.$$

After having obtained COL, one needs to compute $\mathcal{U}_B^{-1} COL$ in order to pivote the new column into the basis. Optimality in the master problem is reached when the objective function of the subproblem is greater than or equal to zero, as this means, that no non-basis variable has negative reduced costs.

3.2.1 Network Flow Based Subproblem Formulation

Not much has been said so far in finding a classification of the subproblem in terms of existing mathematical models. This will now be changed. The given subproblem, i.e. finding a shape matrix with minimal reduced costs, can be modelled as a minimum cost network flow problem with additional constraints. As the column generation problem creates a single shape matrix, tongue-and-groove effects need not to be discussed in this framework. Additionally, we will neglect the interleaf motion constraints, but will add it to the set of network flow constraints later again.

Figure 3.1: *A single row of the MLC represented by main vertices and boxes.*

Each channel of the MLC can be modelled by the set of vertices given in Figure 3.1.

Each main vertex corresponds to a location where the leaf-ends can be positioned. Each of the boxes corresponds to a cell in a row of the MLC grid that is to be irradiated. So the underlying intensity matrix for Figure 3.1 has 4 columns.
We will introduce two more main vertices in every channel: a left and a right starting vertex, as in Figure 3.2 and we connect each of them to their neighbouring main vertex by an arc. The lower bound for each of these arcs equals one - this is to ensure, that neither two left nor two right leaves are modelled, but both a leaf and a right leaf in each channel.

Figure 3.2: *Starting vertices have been added to the very right and left, respectively.*

For each cell, i.e. box, we introduce two mini vertices and arcs connecting the two mini vertices to each other and to the main vertices. This results in the following graph

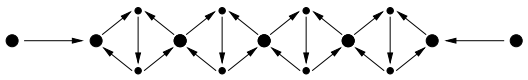


Figure 3.3: *Graph of a row.*

Each main vertex is then connected to the left and the right most starting vertex in the channel below. The last row's main vertices get connected to two sink nodes each. Each of these sink nodes has a demand of one unit. The starting vertices in the first row of the resulting MLC graph (cf. Figure 3.4) both have a supply of one unit.

The aim is to look for a flow of value one from each of the starting vertices to a sink node. Thus the overall flow has a value of two. One might wonder how to setup the variables in order to connect the original column generation problem with this flow problem. Due to the upper bound constraint on the dashed arcs, all arcs carry a flow of either zero or one. The variables corresponding to the dashed arcs will be the original variables in the column generation subproblem, namely $y_{i,j} = 1 - COL_{(i-1)*n+j}$. In fact these are the only variables of non-zero cost. The costs of y are given by the dual multipliers $\underline{1}_B \mathcal{U}_B^{-1}$, and the objective to minimize is $-\underline{1}_B \mathcal{U}_B^{-1} y$. So we are looking for a minimum cost flow of two units from the sources to the sinks in the given network structure.

As stated in the beginning of this section interleaf motion constraints have not been handled so far. There is actually no possibility to include the interleaf motion constraints in the network flow based formulation described above. Therefore, two additional constraints are needed, in

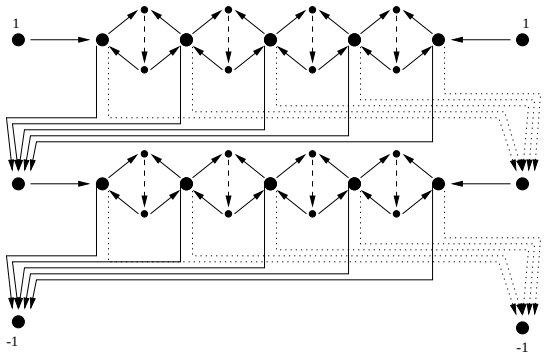


Figure 3.4: *The MLC Graph. Dashed arcs have an upper bound of one.*

order to ensure the removal of interleaf motion. Before stating the new constraints, a detailed overview about the naming of some of the variables is given in Figure 3.5.

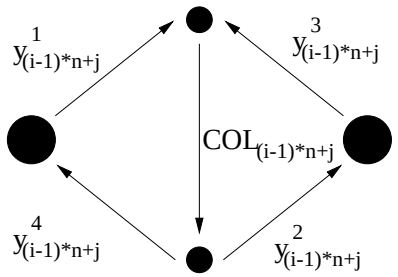


Figure 3.5: *Variables' assignment in channel i, cell j.*

It is now possible to give a criterion which detects and thus can be used to forbid interleaf motion.

Theorem: Interleaf motion does not occur, if and only if the following constraints are met:

$$y^1_{(i-1)*n+j} + y^3_{i*n+j} \quad \leq \quad 1 \quad \forall\, i \in \{1,\ldots,m\},\ j \in \{1,\ldots,n\}$$

Proof: The constraints can be interpreted as follows: if $y^1_{(i-1)*n+j} = 1$ then the left leaf is positioned to the *right* of cell j in channel i. Vice versa, if $y^3_{i*n+j} = 1$ then the right leaf is positioned to the *left* of cell j in the channel below, which implies interleaf motion of the two adjacent leaves. As $y, y^1, y^2, y^3, y^4 \in \{0,1\}^{m*n}$ all other combinations are valid with respect to interleaf motion and it can easily be checked that the constraint evaluates to a value less than or equal to one in these valid cases.

q.e.d.

3.2.2 Tongue-and-Groove Elimination

Reviewing the constraints to exclude the tongue-and-groove effect in the first MIP model given, it can be observed that the number of constraints needed is very large. Especially the fact,

that the constraint has to hold for any two chosen segments increases the number of constraints enormously. Although it seems that the desire to remove tongue-and-groove effects, implies a huge number of constraints, it is actually not the case. A formulation which reduces the number of constraints to $(m-1)*n$ in the master problem is now introduced.

Suppose one is given an $m \times n$ intensity matrix. Each entry is again represented as a box in Figure 3.6. Each of the boxes will either be masked or unmasked in a valid shape matrix. The tongue and groove effect can occur on the gridlines in between two neighbouring cells in adjacent channels. These gridlines are represented by the red lines in Figure 3.6.

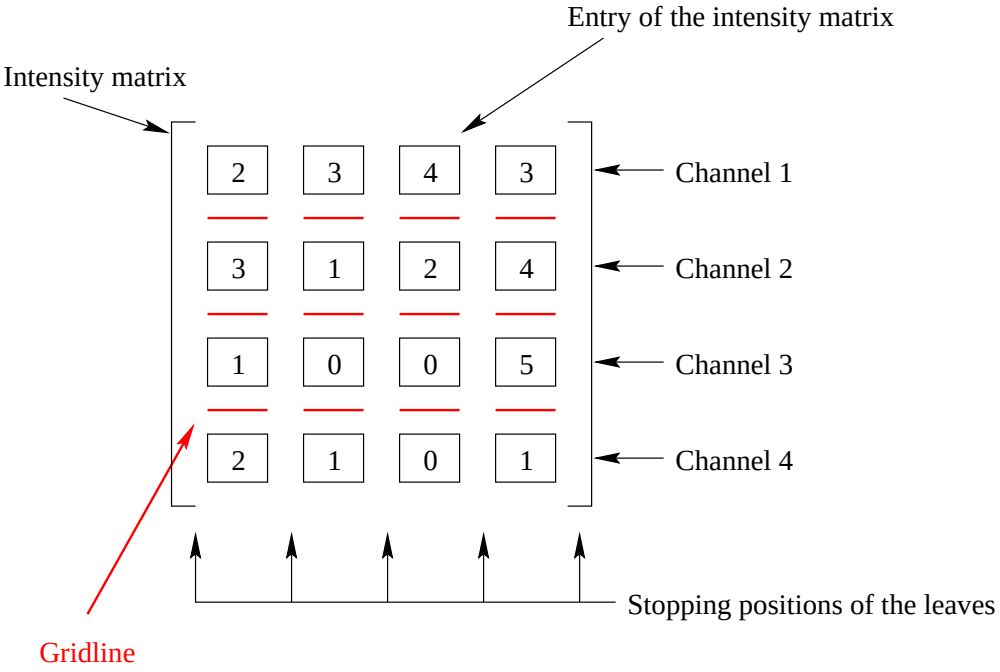


Figure 3.6: *Intensity matrix including the horizontal gridlines.*

By the definition of tongue-and-groove, it is known, that the effect occurs, when the adjacent cells are irradiated alternately. The gridline is irradiated whenever at least one of the adjacent cells is irradiated. Thus, it can be concluded that the tongue-and-groove effect is completely removed, if the gridline is irradiated in as many segments as is the maximal number of intensity units of the adjacent cells. A very small example will help to clarify everything.

Given the intensity matrix $\begin{pmatrix} \boxed{4} \\ — \\ \boxed{3} \end{pmatrix}$ then the maximal intensity requirement is 4 units for the upper cell. Thus the gridline in between these cells shall not be irradiated for more than 4 units. If the decomposition fulfills this requirement, then no tongue-and-groove occurs. The following theorem stresses this fact and even states that the converse is true as well. But before stating the theorem, a short introduction on how to include the new constraints has to be given, as the notation will be changed slightly.

For each of the $(m-1)*n$ horizontal gridlines, we add a constraint in the master problem. The

new matrix is denoted by

$$\hat{\mathcal{U}} := \begin{pmatrix} \mathcal{U} \\ \mathcal{C} \end{pmatrix}, \text{ where } \mathcal{C} \in \{0,1\}^{(m-1)*n \times r}$$

and is defined by

$$\hat{\mathcal{U}}_{(i-1)*n+j+m*n,q} := \mathcal{C}_{(i-1)*n+j,q} := \max\left\{\mathcal{U}_{(i-1)*n+j,q}, \mathcal{U}_{i*n+j,q}\right\}$$

for all $i \in \{1,\ldots,m-1\}, j \in \{1,\ldots,n\}$. Additionally, the right hand side has to be extended as well, and the new right hand side is

$$\hat{\underline{\mathcal{I}}} := \begin{pmatrix} \underline{\mathcal{I}} \\ \underline{\mathcal{R}} \end{pmatrix}, \text{ where } \hat{\underline{\mathcal{I}}}_{(i-1)*n+j+m*n} := \underline{\mathcal{R}}_{(i-1)*n+j,q} := \max\left\{\mathcal{I}_{i-1,j}, \mathcal{I}_{i,j}\right\}$$

where $i \in \{1,\ldots,m-1\}, j \in \{1,\ldots,n\}$. The new master problem is

$$\left(P^{tng}_{Master}\right) \quad \begin{array}{ll} \min & \underline{1}^T\alpha \\ \text{s.t.} & \hat{\mathcal{U}}\alpha = \hat{\underline{\mathcal{I}}} \\ & \alpha \geq 0 \end{array}$$

In the following M will be defined by $M := m * n$.

Theorem: The tongue-and-groove effect occurs if and only if

$$\sum_{q\in Q} \hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q > \max\left\{\mathcal{I}_{i-1,j}, \mathcal{I}_{i,j}\right\}\ \forall\ i \in \{1,\ldots,m-1\},\ j \in \{1,\ldots,n\}$$

Proof: Note that:

1) Q is the set of columns of $\hat{\mathcal{U}}$, that is the set $\{1,\ldots,r\}$.

2) $\sum_{q\in Q} \hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q \geq \max\left\{\mathcal{I}_{i-1,j}, \mathcal{I}_{i,j}\right\}$. Otherwise the intensity profile cannot be met in cell j of channel k , where k is defined such that $\mathcal{I}_{k,j} = \max\left\{\mathcal{I}_{i-1,j}, \mathcal{I}_{i,j}\right\}$ as

$$\sum_{q\in Q} \hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q \geq \sum_{q\in Q} \hat{\mathcal{U}}_{(k-1)*n+j,q}\ \alpha_q.$$

Suppose that

$$\begin{array}{ll} & \sum_{q\in Q} \hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q < \mathcal{I}_{k,j} \\ \Longrightarrow & \sum_{q\in Q} \hat{\mathcal{U}}_{(k-1)*n+j,q}\ \alpha_q < \mathcal{I}_{k,j} \\ \Longrightarrow & \text{the intensity profile in cell } j \text{ of channel } k \text{ is not met} \end{array}$$

3) Remember that α is the solution vector of the master problem.

4) In the following let i and j be arbitrarily chosen such that $i \in \{1,\ldots,m-1\}$, $j \in \{1,\ldots,n\}$.

$''\Longrightarrow''$ Let $\sum_{q\in Q}\hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q \le \max\{I_{i-1,j}, I_{i,j}\}$

$\overset{2)}{\Longrightarrow}\ \sum_{q\in Q}\hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q = \max\{I_{i-1,j}, I_{i,j}\}$

Let k be such that $I_{k,j} = \max\{I_{i-1,j}, I_{i,j}\}$

l be such that $I_{l,j} = \min\{I_{i-1,j}, I_{i,j}\}$

$k \neq l$.

Since $\sum_{q\in Q}\hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q = I_{k,j}$

and $\sum_{q\in Q}\hat{\mathcal{U}}_{(k-1)*n+j,q}\ \alpha_q = I_{k,j}$

and $\hat{\mathcal{U}}_{s,q} \in \{0,1\},\ \hat{\mathcal{U}}_{s+M,q} \in \{0,1\},\ \hat{\mathcal{U}}_{s+M,q} \ge \hat{\mathcal{U}}_{s,q},\ s \in \{1,\dots,M\},\ q \in Q$

$\Longrightarrow \hat{\mathcal{U}}_{(k-1)*n+j,q} = \hat{\mathcal{U}}_{(i-1)*n+j+M,q} \quad \forall\ q \in Q,\ \alpha_q > 0.$

Additionally $\hat{\mathcal{U}}_{(i-1)*n+j+M,q} = \max\left\{\hat{\mathcal{U}}_{(i-1)*n+j,q}, \hat{\mathcal{U}}_{i*n+j,q}\right\}$

$\Longrightarrow \hat{\mathcal{U}}_{(k-1)*n+j,q} \ge \hat{\mathcal{U}}_{(l-1)*n+j,q} \quad \forall\ q \in Q,\ \alpha_q > 0.$

$\Longrightarrow$ No tongue and groove effect can occur.

$''\Longleftarrow''$ Let $\sum_{q\in Q}\hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q > \max\{I_{i-1,j}, I_{i,j}\}$

and let k and l be as above.

Case 1: $I_{k,j} > I_{l,j}$

$\Longrightarrow\ \exists\ q \in Q : \hat{\mathcal{U}}_{(k-1)*n+j,q} > \hat{\mathcal{U}}_{(l-1)*n+j,q},\ \alpha_q > 0$

$\Longrightarrow\ \exists\ q \in Q : \hat{\mathcal{U}}_{(k-1)*n+j,q} = 1,\ \hat{\mathcal{U}}_{(l-1)*n+j,q} = 0,\ \alpha_q > 0$

Since $\sum_{q\in Q}\hat{\mathcal{U}}_{(i-1)*n+j+M,q}\ \alpha_q > \sum_{q\in Q}\hat{\mathcal{U}}_{(k-1)*n+j,q}\ \alpha_q$

$\underbrace{\Longrightarrow}_{\text{Def. of } \hat{\mathcal{U}}_{(i-1)*n+j+M,q}}\ \exists\ q^* \in Q : \hat{\mathcal{U}}_{(l-1)*n+j,q^*} > \hat{\mathcal{U}}_{(k-1)*n+j,q^*},\ \alpha_{q^*} > 0$

$\Longrightarrow\ \exists\ q^* \in Q : \hat{\mathcal{U}}_{(l-1)*n+j,q^*} = 1,\ \hat{\mathcal{U}}_{(k-1)*n+j,q^*} = 0,\ \alpha_{q^*} > 0$

This implies that tongue-and-groove effects occur.

Case 2: $I_{k,j} = I_{l,j}$

$\exists\ q \in Q : \hat{\mathcal{U}}_{(l-1)*n+j,q} > \hat{\mathcal{U}}_{(k-1)*n+j,q},\ \alpha_q > 0$

Suppose $\nexists\ q^* \in Q :\ \hat{\mathcal{U}}_{(k-1)*n+j,q^*} > \hat{\mathcal{U}}_{(l-1)*n+j,q^*},\ \alpha_{q^*} > 0$

$\Longrightarrow \hat{\mathcal{U}}_{(l-1)*n+j,q} \ge \hat{\mathcal{U}}_{(k-1)*n+j,q},\ \forall\ q \in Q,\ \alpha_q > 0$

and $\hat{\mathcal{U}}_{(l-1)*n+j,q} > \hat{\mathcal{U}}_{(k-1)*n+j,q}$ for at least one $q \in Q$ with $\alpha_q > 0$.

Hence it follows that $I_{l,j} > I_{k,j}$.

A contradiction, which implies that the assumption was wrong

and thus tongue-and-groove effects occur.

q.e.d.

As the master problem has changed, it is necessary to change the column generation procedure for obvious reasons. It is no longer valid to simply produce shape matrices written as a column, but extending these columns by the coefficients for the new tongue-and-groove removing constraints will resolve this problem. According to the old definition of the subproblem we have to define a new set of variables, which correspond to the new constraints of the master problem. These new variables will be called $\hat{y} \in \{0,1\}^{(m-1)\times n}$ according to the variables $y \in \{0,1\}^{m\times n}$ of the old constraints of the subproblem. The new subproblem will then be as follows:

$$
\left(P_{CG}^{tng}\right)
\begin{array}{lll}
\min & 1 - \underline{1}_B^T \hat{\mathcal{U}}_B^{-1} \begin{pmatrix} y \\ \hat{y} \end{pmatrix} & \\
\text{s.t.} & \text{Constraints of } (P_{CG}) \text{ and} & \\
& \hat{y}_{i,j} \geq y_{i,j} & \forall\, i \in \{1,\ldots,m-1\},\ j \in \{1,\ldots,n\} \\
& \hat{y}_{i,j} \geq y_{i+1,j} & \forall\, i \in \{1,\ldots,m-1\},\ j \in \{1,\ldots,n\} \\
& \hat{y}_{i,j} \leq y_{i,j} + y_{i+1,j} & \forall\, i \in \{1,\ldots,m-1\},\ j \in \{1,\ldots,n\} \\
& \hat{y}_{i,j} \in \{0,1\} & \forall\, i \in \{1,\ldots,m-1\},\ j \in \{1,\ldots,n\}
\end{array}
$$

3.2.3 Modelling The Setup Time

Given the intensity matrix $\mathcal{I}$ we scale it down using the largest of its entries and obtain a new matrix $\mathcal{I}^s$ which is hence defined by the relation

$$\mathcal{I}^s_{ij} := \frac{\mathcal{I}_{ij}}{\max_{k,l}\{\mathcal{I}_{k,l}\}}.$$

Using this normalization, we will decompose the new matrix and use the fact, that each relative beam-on time coefficient α_q satisfies

$$0 \leq \alpha_q \leq 1.$$

This will enable us to introduce fixed costs (i.e. extra time) for each segment being used – regardless for how long – , namely by introducing new binary variables $f \in \{0,1\}^r$ which shall be defined as follows:

$$f_q := \begin{cases} 1 & \text{if shape matrix } q \text{ is used in the decomposition} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, q \in \{1,\ldots,r\}.$$

This definition can be realized by adding the following constraints to the master problem

$$\left.\begin{array}{rcl} f_q & \geq & \alpha_q \\ f_q & \in & \{0,1\} \end{array}\right\} \ \forall\, q \in \{1,\ldots,r\}$$

or equivalently using additional surplus variables f^{sp}

$$\left.\begin{array}{rcl} f_q - f_q^{sp} & = & \alpha_q \\ f_q & \in & \{0,1\} \\ f_q^{sp} & \geq & 0 \end{array}\right\} \ \forall\, q \in \{1,\ldots,r\}.$$

The goal which was aimed at by introducing these new variables was a better representation of the total delivery time by the objective function of the master problem , which will be changed to the following

$$\min \left(\underline{1}^T f\right) * OH_{VR} + \left(\underline{1}^T \alpha\right) * \max_{k,l} \{\mathcal{I}_{k,l}\} * C_{D_R}.$$

It is obvious, that due to the changes in the master problem, the column generation routine will have to be changed appropriately. The new constraints for the existing column can easily be introduced in the current column generation procedure, whereas a second column generation procedure is needed to produce the columns corresponding to the newly defined variables in the case, that no other column with negative reduced cost can be found.

Chapter 4

Computational Complexity

An interesting topic which has not been discussed so far is the complexity of an algorithm to solve the underlying problem. As it is very hard to discuss complexity issues on a general mixed integer programming formulation, as given in (MIP), or on a general column generation approach, as given in P_{Master}, P^{tng}_{Master} together with P_{CG} and P^{tng}_{CG}, respectively, it is of high interest to find a polynomial transformation of the problem into a new formulation, which has already been classified with respect to computational complexity aspects.

As given in the previous section there is a link to a network flow formulation, though not a pure network flow problem since additional constraints had to be added to avoid interleaf motion. On the other hand, the network flow based formulation given beforehand, was only designed to define a single shape matrix, and neither was it proposed to decompose a given intensity profile, nor to handle tongue-and-groove effects. These issues will have to be discussed.

4.1 Theory of Computational Complexity

Given an optimization problem (OP) then the corresponding decision problem (DP) answers the question whether there exists a solution feasible to the constraints of (OP) with an objective value less than or equal to an integer k. If it can be shown that (DP) is $\mathcal{NP}$-complete, then (OP) is $\mathcal{NP}$-hard, as it is at least as difficult to solve as (DP) and it is not known to be a member of $\mathcal{NP}$ [Garey & Johnson (1979)][1]. Suppose there exists a second problem $(OP)_2$, then there are several ways to show that it is $\mathcal{NP}$-hard as well. One way is to find a polynomial Turing Machine reduction from $(OP)_2$ to (OP), which implies the desired property. Another way to prove this, is to show that the decision problem $(DP)_2$ which is derived from $(OP)_2$, i.e. asking whether there exists a solution to $(OP)_2$ with an objective value less than or equal to an integer k, is polynomially transformable to (DP). Then, due to the fact that (DP) is known to be $\mathcal{NP}$-complete, $(DP)_2$ is, too. Thus, $(OP)_2$ – the optimization problem corresponding to $(DP)_2$ – is $\mathcal{NP}$-hard as it is at least as difficult to solve as $(DP)_2$ and it is not known to be a member of $\mathcal{NP}$. In the following, a formulation of the problem is developed which is known to be $\mathcal{NP}$-hard. Given the original problem it will then be shown, that the decision problem associated with the real world problem formulation is polynomially transformable to the other.

[1] It should be taken care of the fact that this thesis cannot cover the complete theory of computational complexity and the interested reader is referred to [Garey & Johnson (1979)].

4.2 A Network Flow Based Formulation of The Problem

4.2.1 Why A Simple Network Won't Work

An intensity profile (e.g. the 1-dimensional profile (1 2 4 2)) can be represented by the net number of L- and R-leaf positions. Suppose the intensity requirement of cell i is 2 units higher than the adjacent cell $i-1$ in a particular channel j. The net number of L-leaf positions at position i is thus 2. The net number of an L-leaf position at a certain position of a channel of the MLC is the net number of segments in which the left leaf has to be positioned at the left border of this cell. In this context *net* means, that in total there may be more segments with the L-leaf positioned at location i, but, in order to meet the intensity requirement, one has to substract the total number of segments in which the opposing right leaf was positioned at this location. A similar observation and description can be given for the net number of R-leaf positions.
In the above example the net number of left and right leaf positions would be as given in Figure 4.1.

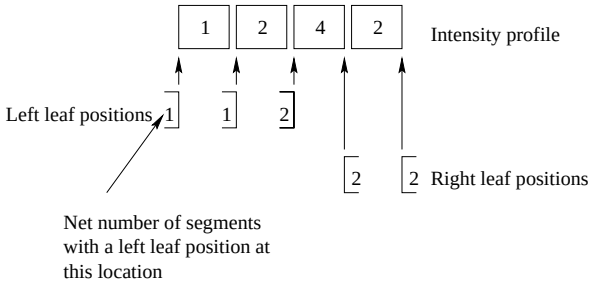


Figure 4.1: *An intensity profile for a single-channel MLC with the net number of L- and R-leaf positions.*

In order to model a flow for a single-channel MLC which constructs a decomposition of the intensity profile, a network as given in Figure 4.2 is constructed:

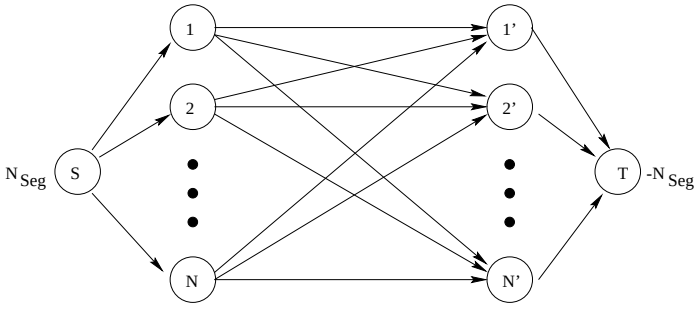


Figure 4.2: *Network for a single segment designed in order to decompose a 1 dimensional intensity profile. A flow from i to j, $i, j \neq S, T$, represents a segment, where the left leaf position is positioned at i and the right leaf is positioned at j.*

In each of these subnetworks one is looking for a flow of N_{Seg} units from S to T such that the sum of all flows from S through i minus the sum of all flows from i' to T equals the net number

of L-leaf positions at i. A similar condition holds for the R-leaf positions. It is important to note that a position has either a positive value for the net number of L-leaf positions or a positive value for the net number of R-leaf positions, but not both! Of course, N_{Seg} is the number of segments needed. A careful observation shows that either one has to compute an upper bound on the number of segments or no optimization will be necessary. In the one-dimensional case there is actually no need for an optimization: it is known beforehand that the number of segments in a single-channel MLC equals to the sum of the net number of L-leaf positions, which is of course the same as the sum of the net number of R-leaf positions [Webb (1998a)]. This is very easy to verify as in a single-channel MLC there is basically no restriction on the leaf setting, save the fact, that the left leaf has to be positioned to the left of the right leaf. Thus we can decompose the required intensity profile by a simple sweep along the channel, i.e. one initially places the leaves to their left most position and in each setup the leaves either keep their positions or will be moved to the right. But the model is designed to solve the two dimensional case!

Increasing the supply and demand equally to a value k, i.e. an upper bound on the number of needed segments for the decomposition, and pricing the arcs as follows

$$c_{i,j} = \begin{cases} 0 & \text{if } i = S \text{ or } j = T \\ 1 & \text{if } i \neq S \text{ and } j \neq T \end{cases}$$

will allow to construct a decomposition without the knowledge of N_{Seg}. In this case one is looking for a minimum cost flow of k units from S to T. The problem can then be mathematically stated as follows:

$$\begin{array}{lllll} \min & \displaystyle\sum_{i,j,i\neq S,j\neq T} c_{i,j}x_{i,j} & & & \\ \text{s.t.} & x_{S,i} - x_{i,T} & = & N_i^L & \forall\ i : N_i^L \geq 0 \\ & x_{i,T} - x_{S,i} & = & N_i^R & \forall\ i : N_i^R \geq 0 \\ & \displaystyle\sum_{i\neq S} x_{S,i} & = & \displaystyle\sum_{j\neq T} x_{j,T} & \\ & x_{S,i} & = & \displaystyle\sum_{j} x_{i,j} & \forall\ i \\ & x_{j,T} & = & \displaystyle\sum_{i} x_{i,j} & \forall\ j \\ & x_{i,j} & \geq & 0 & \forall\ i,j \end{array}$$

But, extending this model to a multi-channel MLC results in a nonlinear optimization problem, which is definitely not desired. Nevertheless, the explanation of how the non-linearity comes into play will be given. Dealing with a multi-channel MLC from now on, the network has to be extended. We basically copy the given network m times, where m is again the number of channels of the MLC. Each of these copies represents one channel of the MLC. Then the demand and the supply of each source and sink is set as in the single-channel MLC to a value H which is an upper bound on the number of segments needed. It should be stressed, that this is not the maximal value of all single channel upper bounds, but it can be as high as the sum of all L-leaf positions in the MLC. For the multi-channel MLC it is important to know when certain positions are paired, thus, we can not allow a flow of more than one unit along each arc. On the other hand it might be necessary that certain pairings are realized in more than one segment of the decomposition. Thus, each arc from a left leaf position i to a right leaf position j has to be

copied H times. Each of these arcs gets an upper bound of one. The arc set of the C-th copy of the network will be named A^C. We define the flow variables for all channels $C \in \{1, \ldots, m\}$ and for all segments $t \in \mathcal{T} := \{1, \ldots, H\}$ to be

$$x_{i,j}^C(t) := \begin{cases} 1 & \text{if there is a flow of 1 unit from } i \text{ to } j \text{ in channel } C \text{ in segment t} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, (i,j) \in A^C$$

and penalty costs for the removal of the interleaf motion of leaves of channels $C \in \{1, \ldots, m-1\}$ as follows

$$\hat{c}_{i,j,k,l}^{C,C+1} := \begin{cases} 1, \text{if } i > l \text{ or } j < k \\ 0, \text{otherwise} \end{cases} \qquad \forall\, (i,j) \in A^C,\ (k,l) \in A^{C+1},\ i,k \neq S,\ j,l \neq T$$

The underlying networks for a 2-channel MLC with $H = 3$ look as given in Figure 4.3.

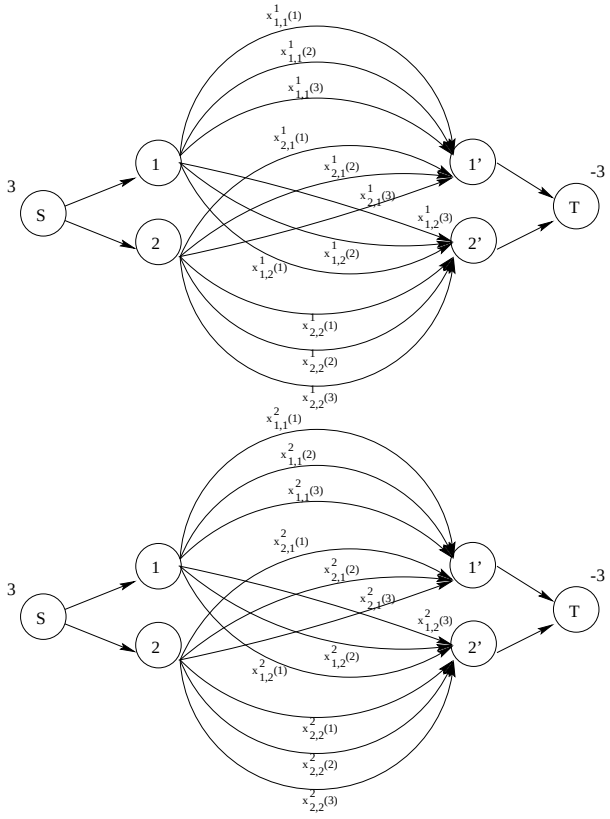


Figure 4.3: *Networks for a 2-channel MLC with two leaf stopping positions and $H = 3$.*

In order to forbid interleaf motion and to minimize the number of segments one is then looking

for the lexicographical minimum of

$$\left(\begin{array}{cc} \sum\limits_{\substack{t \in \mathcal{T} \\ C \in \{1,\ldots,m-1\} \\ (i,j) \in A^C, i \neq S, j \neq T \\ (k,l) \in A^{C+1}, k \neq S, l \neq T}} & \hat{c}_{i,j,k,l}^{C,C+1}\ x_{i,j}^{C}(t)\ x_{k,l}^{C+1}(t) \\ \sum\limits_{\substack{t \in \mathcal{T} \\ C \in \{1,\ldots,m\} \\ (i,j) \in A^C, i \neq S, j \neq T}} & c_{i,j}\ x_{i,j}^{C}(t) \end{array} \right)$$

The first objective reduces the number of interleaf motions between adjacent channels to zero. The second component will then be minimized among all solutions which satisfy no interleaf motion.

4.2.2 Overcoming Nonlinearity

Nonlinearity is a bit of an undesired feature when dealing with network flow formulations with additional constraints. On the other hand it can easily be seen, that the "most" information is needed by the tongue-and-groove constraints to detect a single violation. The required information is:

- L-leaf position i in row C in segment t
- R-leaf position j in row C in segment t

- L-leaf position $\hat{i}$ in row $C+1$ in segment t
- R-leaf position $\hat{j}$ in row $C+1$ in segment t

- L-leaf position k in row C in segment $t^*,\ t < t^*$
- R-leaf position l in row C in segment $t^*,\ t < t^*$

- L-leaf position $\hat{k}$ in row $C+1$ in segment $t^*,\ t < t^*$
- R-leaf position $\hat{l}$ in row $C+1$ in segment $t^*,\ t < t^*$

Creating a new network which overcomes the nonlinear aspects of the previously defined network consists of vertices, which will be named as follows: Thus, we introduce vertices of the form

$$\boxed{i, j, k, l, C, t, t^*}$$

This vertex represents the following state of channel C and segment t:

- the left leaf is positioned to the left of cell i,

- the right leaf is positioned to the left of cell j,

and in segment t^*:

- the left leaf is positioned to the left of cell k,
- the right leaf is positioned to the left of cell l.

In order to decompose the intensity profile, one is looking for a matching of vertices as follows:

A Exactly one vertex

$$\boxed{i, j, k, l, C, t, t^*}$$

of every channel C and a segment pairing t, t^* is matched with another vertex

$$\boxed{\hat{i}, \hat{j}, \hat{k}, \hat{l}, C+1, t, t^*}$$

in channel $C+1$. I.e. as soon as this matching has been determined, we have fixed the position of the left and right leaves in channels C and $C+1$, but only in the segments t and t^*. The latter vertex will then be matched to another vertex

$$\boxed{\hat{\hat{i}}, \hat{\hat{j}}, \hat{\hat{k}}, \hat{\hat{l}}, C+2, t, t^*}$$

in channel $C+2$, if $C+1$ was not the last channel of the MLC. Having reached the last channel of the MLC one has determined all L- and R-leaf positions in all channels in the segments t and t^*. In order to reduce the tongue-and-groove effect to zero, it is necessary to forbid tongue-and-groove effects between all possible pairings of segments. Thus, the same procedure has to be repeated for all other segments $\hat{t},\ \hat{t}^*$ in order to penalize certain matchings. Here we face difficulties in keeping the consistency of the solution: suppose we have already determined the vertex-matching in the segment pairing $t,\ t^*$. Then for all other segment pairings $t,\ \hat{t}^*$ and $\hat{\hat{t}}^*,\ t$ we have to ensure that the matching will be such that a leaf-pair placement decision which has been made in the pairing $t,\ t^*$ will be the same for segment t in the pairing of segments $t,\ \hat{t}^*$ and $\hat{\hat{t}}^*,\ t$. Of course, the same difficulty will be observed for t^*. Thus, we will need to introduce additional matchings which will assure this property. In the following, a more explicit list is given of what the implications of the matching of vertex

$$\boxed{i, j, k, l, C, t, t^*}$$

to another vertex

$$\boxed{\hat{i}, \hat{j}, \hat{k}, \hat{l}, C+1, t, t^*}$$

are.

B In another segment combination $t,\ t^{**},\ t^* \neq t^{**}$, consistency requires that vertices representing channel C leaf settings are of the form

$$\boxed{o, p, q, r, C, t, t^{**}}$$

where $o = i,\ p = j,\ q$ and r can be chosen arbitrarily.

C On the other hand, if the segments $\bar{t}$ and t are combined then the vertices representing the leaf setting in channel C have to be of the form

$$\boxed{o, p, q, r, C, \bar{t}, t}$$

where $q = i,\ r = j$.

D In the combination of segments $\tilde{t}(\neq t)$ and t^* vertices representing channel C leaf settings are of the form

$$\boxed{o, p, q, r, C, \tilde{t}, t^*}$$

where $q = k,\ r = l$.

E Similarly, when combining the segments t^* and $\hat{t}$ the matched vertices representing channel C have to be of the form

$$\boxed{o, p, q, r, C, t^*, \hat{t}}$$

where $o = k,\ p = l$.

Additionally, the number of matched vertices with fixed channel C and segments t and t^* is exactly one. Otherwise the model would not be consistent, as neither zero nor more than one leaf pair pairing per segment and channel are senseful.

Realizing A For fixed segments t and t^*, a source and a sink with a supply and a demand of one, respectively, will be introduced. The source will be connected to all nodes with $C = 1$ of the combination t and t^*. All vertices of this segment combination of a particular channel C are connected with a directed arc to all nodes in that combination and channel $C + 1$. The last "row" of vertices, i.e. vertices in this segment combination with $C = m$ are connected to the sink. A sketch of the network is shown in Figure 4.4. The flow of 1 unit from S to T in this network will be called the in-segment-combination flow.

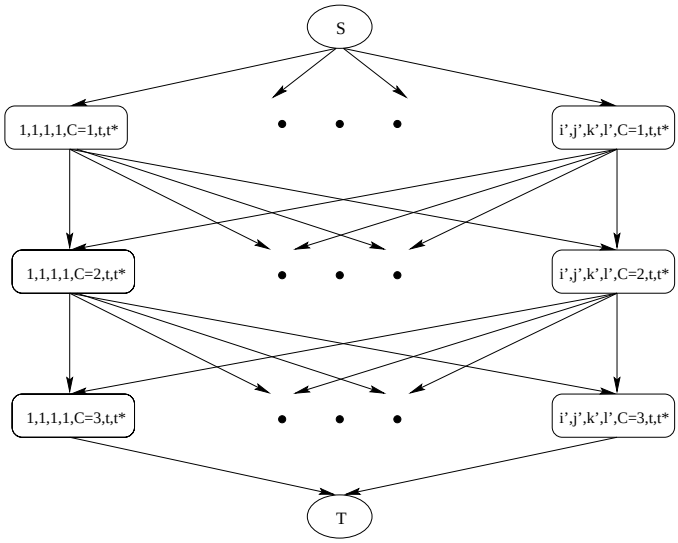


Figure 4.4: *A subnetwork.*

Before going into a more detailed explanation on how B, C, and D are realized we outline the idea. Figure 4.5 sketches the whole network, where each box represents a subnetwork as described above and in Figure 4.4. In order to preserve consistency, it was observed, that a segment combination $t,\ t^*$ has in some way to be linked with all other segment combinations $t,\ \tilde{t}$. Here it is important to realize, that a linkage is transitive, i.e. when $t,\ t^*$ has been linked to $t,\ \tilde{t}$ and the latter to $t,\ \hat{t}$, then a direct link between t, t^* and $t,\ \hat{t}$ is not necessary for obvious reasons. Hence, Figure 4.5 displays all relevant linkages of segment combinations.

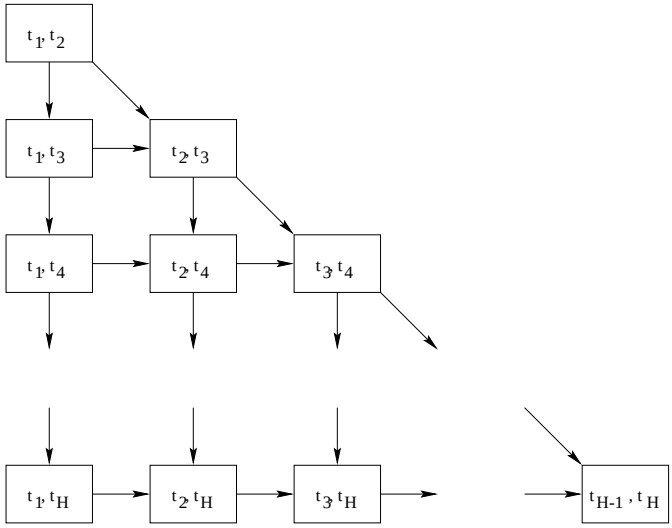


Figure 4.5: *The linkages.*

Realizing B and C The realization of the requirements of B stated in the list above will be done using the vertical connections in Figure 4.5. Reviewing the consistency requirement in B, this connection will ensure that a leaf setting decision for t made in the segment pair $t,\ t^*$ will still hold in any segment combination $t,\ \tilde{t}$. As this has to be done for each channel, $H-2$ additional sources and sinks, each with a supply and a demand of m units each, respectively, will be introduced. The sources will be named $S, t_p, \downarrow$, where $p < H-1$. It will be connected to all vertices of the segment pair t_p, t_{p+1}. In addition to this, the following arc connections will be created: each vertex $i, j, k, l, C, t_p, t_{p+k}$ is connected to all vertices of the form $i, j, q, r, C, t_p, t_{p+k+1}$ if $t_{p+k} \neq t_H$. In the case that $t_{p+k} = t_H$ all vertices are connected to the sink, named $T, t_p, \downarrow$.

Realizing D Here additional sources, sinks and arcs are constructed in order to meet the constraints of D. Therefore $H-2$ sources and sinks, each with a supply and a demand of m units, respectively, are introduced. They will be named $S, t_p, \rightarrow$ and $T, t_p, \rightarrow$, respectively, where $p \geq 3$. Each source $S, t_p, \rightarrow$ will be connected to all vertices of the segment combination t_1, t_p, where $p \geq 3$. Each vertex of the form $i, j, k, l, C, t_{p-k}, t_p$ will be connected to all vertices $q, r, k, l, C, t_{p-k+1}, t_p$, where $p \geq 3$, if $p-k < p-1$. Otherwise it will be connected to the sink $T, t_p, \rightarrow$.

Realizing E This section discusses the technique how consistency is preserved with respect to E in the list above. Therefore we introduce a single source and a single sink, named $S, \searrow$ and

$T, \searrow$, respectively, where the source has again a supply of m units and the sink a demand of m units. Various arcs are added similarly as it was done in the previous paragraph. The source gets connected to all vertices of the segment t_1, t_2. Each vertex of the form $i, j, k, l, C, t_p, t_{p+1}$ will be connected to all vertices $k, l, q, r, C, t_{p+1}, t_{p+2}$, if $p+1 < H$. Otherwise, these vertices are connected to the sink $T, \searrow$.

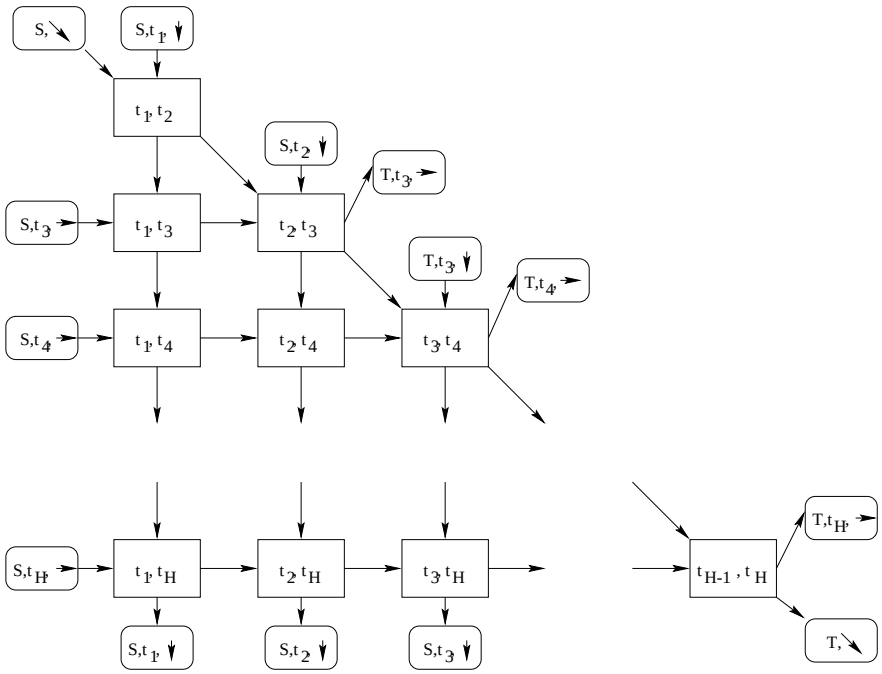


Figure 4.6: *The linkages and the additional sources and sinks.*

In order to ensure consistency of the model the following has to hold: for fixed values of t, t^* and C there is exactly one vertex through which flow passes.

Definition: The flow variables will be named x_{tv}^{hv}, where tv is the tail vertex of the arc, e.g. $i, j, k, l, C, t_p, t_{p+1}$, and hv is the head vertex of the arc, e.g. $i, j, q, r, C, t_p, t_{p+2}$, and of course $x_{tv}^{hv} \in \{0, 1\}$.

Definition: Denote with $\mathcal{L}$ the set of all admissible leaf setting combinations i, j, k, l, where i is the left leaf's position in channel C in segment t, j the opposing right leaf's position, k the left leaf's position in channel C in segment t^* and where l is the opposing right leaf's position. A quadrupel i, j, k, l is admissible, if $i \leq j$ and $k \leq l$.

Penalizing arcs which yield tongue-and-groove, penalizing all arcs connected to vertices which yield either interleaf motion or represent non-physical leaf settings, and setting all other cost coefficients in the network to zero, then the following theorem holds:

Theorem: A flow satisfying the demands in the network described above of zero costs with the additional constraints:

- for all vertices but one with fixed t, t^* and C there is no flow going in or out, which means

that

$$card\left(\left\{(i,j,k,l) : x_{i,j,k,l,C,t,t^*}^{o,p,q,r,\tilde{C},\tilde{t},\tilde{t^*}} > 0 \text{ for any } (o,p,q,r) \in \mathcal{L}, \tilde{C}, \tilde{t}, \tilde{t^*}\right\}\right) = 1 \ \forall \ (i,j,k,l) \in \mathcal{L}$$

- the required net number of L-leaf positions at position a ($N_{a,C}^L$) given by the intensity profile of channel C are met for all those positions a and channels C with $N_{a,C}^L \geq 0$, i.e.

$$\begin{aligned}
\sum_{(i,j,k,l)\in\mathcal{L}:i=a} x_{S,t_p,t_{p+1}}^{(i,j,k,l),C=1,t_p,t_{p+1}} - \sum_{(o,p,q,r)\in\mathcal{L}:p=a} x_{S,t_p,t_{p+1}}^{(o,p,q,r),C=1,t_p,t_{p+1}} &= N_{a,C=1}^L, \ p < H-1 \\
\sum_{(o,p,q,r)\in\mathcal{L}:o=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_p,t_{p+1}}^{(o,p,q,r),C+1,t_p,t_{p+1}} - \\
\sum_{(o,p,q,r)\in\mathcal{L}:p=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_p,t_{p+1}}^{(o,p,q,r),C+1,t_p,t_{p+1}} &= N_{a,C}^L, \ p < H-1, \\
&\quad C = 1,\ldots,m-1 \\
\sum_{(i,j,k,l)\in\mathcal{L}:i=a} x_{(i,j,k,l),C=m,t_p,t_{p+1}}^{T,t_p,t_{p+1}} - \sum_{(o,p,q,r)\in\mathcal{L}:p=a} x_{(o,p,q,r),C=m,t_p,t_{p+1}}^{T,t_p,t_{p+1}} &= N_{a,C=m}^L, \ p < H-1 \\
\sum_{(i,j,k,l)\in\mathcal{L}:k=a} x_{S,t_{H-1},t_H}^{(i,j,k,l),C=1,t_{H-1},t_H} - \sum_{(o,p,q,r)\in\mathcal{L}:r=a} x_{S,t_{H-1},t_H}^{(o,p,q,r),C=1,t_{H-1},t_H} &= N_{a,C=1}^L \\
\sum_{(o,p,q,r)\in\mathcal{L}:q=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_{H-1},t_H}^{(o,p,q,r),C+1,t_{H-1},t_H} - \\
\sum_{(o,p,q,r)\in\mathcal{L}:r=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_{H-1},t_H}^{(o,p,q,r),C+1,t_{H-1},t_H} &= N_{a,C}^L \\
&\quad C = 1,\ldots,m-1 \\
\sum_{(i,j,k,l)\in\mathcal{L}:k=a} x_{(i,j,k,l),C=m,t_{H-1},t_H}^{T,t_{H-1},t_H} - \sum_{(o,p,q,r)\in\mathcal{L}:r=a} x_{(o,p,q,r),C=m,t_{H-1},t_H}^{T,t_{H-1},t_H} &= N_{a,C=m}^L
\end{aligned}$$

- A similar set of equations can easily be derived in order to meet the required number of R-leaf positions of the intensity map at position a ($N_{a,C}^R$), where $N_{a,C}^R \geq 0$:

$$\begin{aligned}
\sum_{(i,j,k,l)\in\mathcal{L}:j=a} x_{S,t_p,t_{p+1}}^{(i,j,k,l),C=1,t_p,t_{p+1}} - \sum_{(o,p,q,r)\in\mathcal{L}:o=a} x_{S,t_p,t_{p+1}}^{(o,p,q,r),C=1,t_p,t_{p+1}} &= N_{a,C=1}^R, \ p < H-1 \\
\sum_{(o,p,q,r)\in\mathcal{L}:p=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_p,t_{p+1}}^{(o,p,q,r),C+1,t_p,t_{p+1}} - \\
\sum_{(o,p,q,r)\in\mathcal{L}:o=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_p,t_{p+1}}^{(o,p,q,r),C+1,t_p,t_{p+1}} &= N_{a,C}^R, \ p < H-1 \\
&\quad C = 1,\ldots,m-1 \\
\sum_{(i,j,k,l)\in\mathcal{L}:j=a} x_{(i,j,k,l),C=m,t_p,t_{p+1}}^{T,t_p,t_{p+1}} - \sum_{(o,p,q,r)\in\mathcal{L}:o=a} x_{(o,p,q,r),C=m,t_p,t_{p+1}}^{T,t_p,t_{p+1}} &= N_{a,C=m}^R, \ p < H-1 \\
\sum_{(i,j,k,l)\in\mathcal{L}:l=a} x_{S,t_{H-1},t_H}^{(i,j,k,l),C=1,t_{H-1},t_H} - \sum_{(o,p,q,r)\in\mathcal{L}:q=a} x_{S,t_{H-1},t_H}^{(o,p,q,r),C=1,t_{H-1},t_H} &= N_{a,C=1}^R \\
\sum_{(o,p,q,r)\in\mathcal{L}:r=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_{H-1},t_H}^{(o,p,q,r),C+1,t_{H-1},t_H} - \\
\sum_{(o,p,q,r)\in\mathcal{L}:q=a} \sum_{(i,j,k,l)\in\mathcal{L}} x_{(i,j,k,l),C,t_{H-1},t_H}^{(o,p,q,r),C+1,t_{H-1},t_H} &= N_{a,C}^R \\
&\quad C = 1,\ldots,m-1
\end{aligned}$$

$$\sum_{(i,j,k,l)\in\mathcal{L}:l=a} x^{T,t_{H-1},t_H}_{(i,j,k,l),C=m,t_{H-1},t_H} - \sum_{(o,p,q,r)\in\mathcal{L}:q=a} x^{T,t_{H-1},t_H}_{(o,p,q,r),C=m,t_{H-1},t_H} = N^R_{a,C=m}$$

represents a decomposition of the intensity profile into shape matrices, which are physically deliverable, and do neither violate tongue-and-groove nor interleaf motion constraints. The decomposition's shape matrices can be constructed by setting

$$M^p_{C,i} := \begin{cases} 1 & \text{if the flow through vertex } i,j,k,l,C,t_p,t_{p+1} \text{ with } j \neq i \text{ is greater than zero} \\ 1 & \text{if the flow through vertex } j,i,k,l,C,t_p,t_{p+1} \text{ with } i \neq j \text{ is greater than zero} \\ 0 & \text{otherwise} \end{cases}$$

and by setting all entries $M^p_{C,q}$ to 1, if there exist indices k and l such that $M^p_{C,k} = M^p_{C,l} = 1$ and $k < q < l$.

Proof: By construction of the network.

An overview of the cost structure Summarizing penalty costs, the following list is given to clarify which arcs get which costs. First of all, costs of an arc are either zero or one. And, we are looking for a flow of zero costs, such that the flow through a vertex in the segment combination

- $(t_1,\ t_2)$ and $(t_{H-1},\ t_H)$ consists of either zero or three units;
- $(t_p,\ t_{p+1})$, where $p \neq 1, H-1$, consists of either zero or four units;
- which does not belong to the set described by the two items above, consists of either zero or three units.

The cost strucure is as follows:

- All arcs which are connecting two vertices in different segment combinations have zero costs.
- All arcs which connect the sink or the source have zero costs.
- For the remaining arcs, the following rule holds: if a vertex i,j,k,l,C,t,t^* is connected to a vertex $o,p,q,r,C+1,t,t^*$ with
  - $i > p$, $j < o$, $k > r$ or $l < q$, then we assign a cost factor of one to this connection, as this connection would imply interleaf motion.
  - the existence of an integer c such that one of the tongue-and-groove implying condition holds, stated in the table below, we assign a cost factor of one for this connection. If for the two setups i,j,k,l and o,p,q,r any of the following conditions holds, then let this combination be an element of the set $\mathcal{L}_{tng}$.

Condition	cell c in channel C in segment t	cell c in channel $C+1$ in segment t	cell c in channel C in segment t^*	cell c in channel $C+1$ in segment t^*
at t: $c \geq i,\ c < o,\ c < j$ at t^*: $c < k,\ c \geq q,\ c < r$	unmasked	blocked by L-l.	blocked by L-l.	unmasked
at t: $c \geq j,\ c < p,\ c \geq o$ at t^*: $c < l,\ c \geq r,\ c \geq k$	blocked by R-l.	unmasked	unmasked	blocked by R-l.
at t: $c \geq i,\ c < j,\ c < o$ at t^*: $c \geq l,\ c \geq q,\ c < r$	unmasked	blocked by L-l.	blocked by R-l.	unmasked
at t: $c \geq i,\ c < j,\ c \geq p$ at t^*: $c \geq l,\ c \geq q,\ c < r$	unmasked	blocked by R-l.	blocked by R-l.	unmasked
at t: $c < i,\ c \geq o,\ c < p$ at t^*: $c \geq k,\ c < l,\ c < q$	blocked by L-l.	unmasked	unmasked	blocked by L-l.
at t: $c < i,\ c \geq o,\ c < p$ at t^*: $c \geq k,\ c < l,\ c \geq r$	blocked by L-l.	unmasked	unmasked	blocked by R-l.
at t: $c \geq j,\ c < p,\ c \geq o$ at t^*: $c \geq k,\ c < l,\ c < q$	blocked by R-l.	unmasked	unmasked	blocked by L-l.
at t: $c \geq i,\ c < j,\ c \geq p$ at t^*: $c < i,\ c \geq q,\ c < r$	unmasked	blocked by R-l.	blocked by L-l.	unmasked

Modelling the Treatment Time As given in [Siochi (1999)] the treatment time decomposes into beam-on time and setup-time, whereas the latter is to adjust the MLC in between two non-identical segments, i.e. shape matrices. This is modelled by introducing further H segment combinations $t_p,\ t_p$, where $p = 1, \ldots, H$. Again, these segment combinations have to be linked to the existing segment combinations in order to ensure the consistency of the model. This is done as given in the layout of Figure 4.7. Therefore the sources $S, t_p, \downarrow$ will no longer be connected to all vertices in the segment combination $t_p,\ t_{p+1}$ but to all vertices of the new subnetwork $t_p,\ t_p$, where $p < H$. Finally, the node i, j, i, j, C, t_p, t_p will be connected to all nodes $i, j, k, l, C, t_p, t_{p+1}$. Note that the newly created segments' vertices are all of the form i, j, i, j, C, t_p, t_p and how the combination $(t_H,\ t_H)$ is introduced. The internal structure of these new subnetworks is slightly different from the existing one's. Again these segment combinations have an internal source and sink of a supply and a demand of one unit each. The internal source is connected to all vertices of the form $i, j, i, j, C = 1, t_p, t_p$. Each node i, j, i, j, C, t_p, t_p is connected to the internal sink with a cost of one, if $i \neq j$. In any other case, this vertex is connected to all vertices $k, l, k, l, C + 1, t_p, t_p$, if C was not the last channel of the MLC. In that case, all vertices are connected to the internal sink, but with zero costs, if $i = j$, and with cost 1 if $i \neq j$. And again, we require, in order to maintain consistency, that there is only a single vertex with a fixed channel C and a segment combination t_p, t_p through which flow is passing. Thus, in each segment the internal flow through a segment combination $t_p,\ t_p$ has either cost 1, when passing through a vertex with different left and right leaf positions, i.e. the p-th shape matrix is not the zero matrix, or has cost 0, when all vertices represent closed leaf pairs, thus this shape matrix corresponds to a completely closed MLC. Summing up all of these costs results in the number of segments used, so this model can be used to compute the beam-on time.

In order to give a complete model for the treatment time, we still have to model a flow, the cost

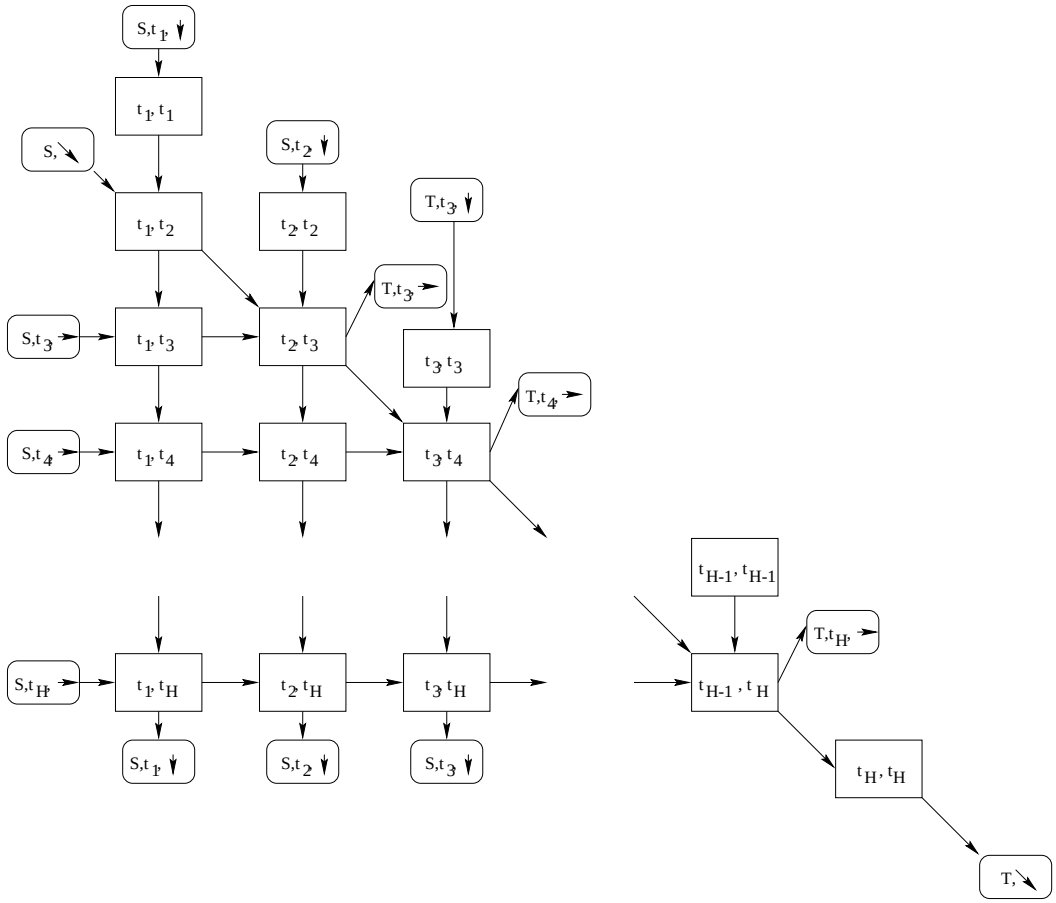


Figure 4.7: *The updated network*

of which represents the setup time in between two nonidentical segments. The idea is similar to the one given above describing the beam-on time. Thus, we increase the supply and the demand of the internal source of the segment combinations t_p, t_{p+1} and introduce additional arcs in this segment combinations as follows: for each vertex of the form $i, j, k, l, C, t_p, t_{p+1}$ we will add an arc to the internal sink if $i \neq k$ or $j \neq l$ with a cost of one and set the lower bound of flow on this arc to one. Additionally, all other arcs departing from this type of vertex are assigned a capacity of one as well. This will force the additional unit flow to be routed to the sink as soon as it arrives at a vertex representing a change in the setup of the MLC. We should take care of the fact that for the last row of the MLC, i.e. $C = m$, we have two parallel arcs from each vertex of the the previously described type to the sink - one with cost equal to one and a lower bound of one (flow defined by $\hat{x}$), and the second does not contribute any cost (flow defined by x). For those vertices with $C = m$ which correspond to leaf pair positions that do not change in the setup procedure between segment t_p and t_{p+1} we will, in accordance with the setup, only have arc connections with zero cost. Of course, consistency requires that there exists exactly one vertex with fixed channel C and segment combination t_p, t_{p+1} through which flow passes. Hence the in-segment-combination flow of a segment combination t_p, t_{p+1} does have a resulting cost of one, if the segments t_p and t_{p+1} are not identical in shape, and zero otherwise. Summing all of these costs, results in the number of different segments.

Finally, the objective is to minimize

$$\sum_{\substack{(i,j,i,j) \in \mathcal{L} : i \neq j \\ C \in \{1,\ldots,m\} \\ t_p \in \{1,\ldots,H\}}} x^{T,t_p,t_p,\downarrow}_{(i,j,i,j),C,t_p,t_p} * C_{D_R} + \sum_{\substack{(i,j,k,l) \in \mathcal{L} : i \neq k \text{ or } j \neq l \\ C \in \{1,\ldots,m\} \\ t_p \in \{1,\ldots,H\}}} \hat{x}^{T,t_p,t_{p+1},\downarrow}_{(i,j,i,j),C,t_p,t_{p+1}} * OH_{VR}$$

under the constraint of zero penalty costs, which can be included in to the objective function using a penalty term.

The constraints are given explicitely in the appendix.

4.3 The Single Source Capacitated Fixed Charge Network Flow Problem

Theorem: The given problem of finding a decomposition of the intensity profile subject to the constraints of the MLC with minimal treatment time can be formulated as a capacitated fixed charge network flow problem.

Proof: In order to transform the network flow based formulation obtained previously, it is necessary to transform the coupling constraints which ensure that only one vertex with fixed t, t^* and C is used into fixed charge constraints by applying fixed costs whenever at least one unit flows "through" such a vertex. Additionally, we need to transform the constraints defining the intensity profile into flow constraints.

- The idea to get rid of the coupling constraints is as follows. Each vertex i, j, k, l, C, t, t^* is split into two vertices defined as described in Figure 4.8 - the arc connecting the two vertices is penalized upon usage with a fixed amount, i.e. no matter how many units pass through this arc the cost will be the same.

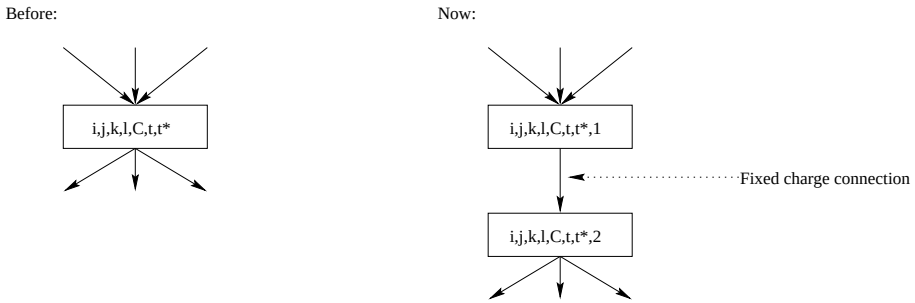


Figure 4.8: *The cellular subnetwork.*

Mathematically speaking we introduce binary variables for these fixed connection and

constraints for all i, j, k, l, C, t, t^* as follows:

$$CAP * x^{i,j,k,l,C,t,t^*,2}_{i,j,k,l,C,t,t^*,1} \geq \sum_{tv \text{ incident with } i,j,k,l,C,t,t^*} x^{i,j,k,l,C,t,t^*,1}_{tv}$$

Clearly, every feasible solution with respect to the constraints of the old formulation will use exatcly one such "fixed charge" connection for a given segment combination t, t^* and a fixed channel C - given high costs for those arcs. On the other hand every feasible solution which satisfies the demand will have at least one non-zero flow along a fixed charge arc per segment combination t, t^* and per channel C. Thus, we are looking for a solution to this fixed charge network flow problem with minimal fixed costs, which then ensures the feasibility to the coupling contraints in the old flow based formulation.

- First of all, a new notation will be introduced. The net number of R- and L-leaf positions of a position j in channel C of the MLC will be given the notation $N^L_{i,C}$ and $N^R_{i,C}$, respectively. Note that not both of these parameters with fixed i and C may have a value which is different from zero. Now, the aim is to model the constraints which ensure the delivery of the intensity profile as a network flow constraint imbedded into the existing network formulation. Therefore, one introduces additional vertices $V^L_{i,C}$ and $V^R_{i,C}$ with a supply of $N^L_{i,C}$ and $N^R_{i,C}$, respectively. Introducing an additional source S_C and connecting it with all vertices $V^L_{i,C}$ and $V^R_{i,C}$ with a supply of zero and linking the latter as follows to the existing network will define the backbone of this new model:

  - Connect $V^L_{i,C}$ to all vertices i, j, i, j, C, t_p, t_p and assign a capacity of one to each arc.
  - Connect $V^R_{i,C}$ to all vertices $j, i, k, l, C, t_p, t_{p+1}$ and assign a capacity of one to each arc.

  The vertices i, j, i, j, C, t_p, t_p in the segment combination t_p, t_p and $k, l, q, r, C, t_p, t_{p+1}$ in the segment combination t_p, t_{p+1} will additionally be connected to a newly defined sink T_C – the demand of which is H – , if $N^L_{i,C} > 0$ and $N^R_{l,C} > 0$, respectively. Those vertices i, j, i, j, C, t_p, t_p for which holds that $N^L_{i,C} = 0$ but $N^R_{i,C} > 0$, will be connected to $V^R_{i,C}$. Similarly $k, l, q, r, C, t_p, t_{p+1}$ will be connected to $V^L_{l,C}$ if $N^R_{l,C} = 0$ and $N^L_{l,C} > 0$. If $N^R_{l,C} = N^L_{l,C} = 0$ then S_C is connected to both $V^L_{l,C}$ and $V^R_{l,C}$, which are in turn connected to vertices l, l, l, l, C, t_p, t_p and $l, l, q, r, C, t_p, t_{p+1}$, repsectively, and the latter are directly connected to the sink T_C. All newly defined arcs have zero cost. The structure of these additional vertices and arcs in the existing network is given in Figure 4.9.

  The definition of these arcs and vertices might cause confusion, thus an interpretation will help. Suppose position i in channel C of the intensity map has a net number of L-leaf position of 3. Thus the supply of the vertex $V^L_{i,C}$ will by definition equal 3. The connection properties of this vertex to the existing vertices will ensure that these 3 units will, on their way towards the sink, pass through existing vertices which represent an MLC shape in channel C such that the left leaf position is at i. Thus, the number of segments with a left leaf positioned at i in channel C is at least the net number of L-leaf positions at this location. This might, as we have seen in the definition of the net number of L- and R-leaf positions, not suffice to deliver the intensity map, since it might happen that the intensity map requires the right leaf of channel C to be placed at position i as well, which implies that at least one more segment is needed with a left leaf positioned at i, in addition to

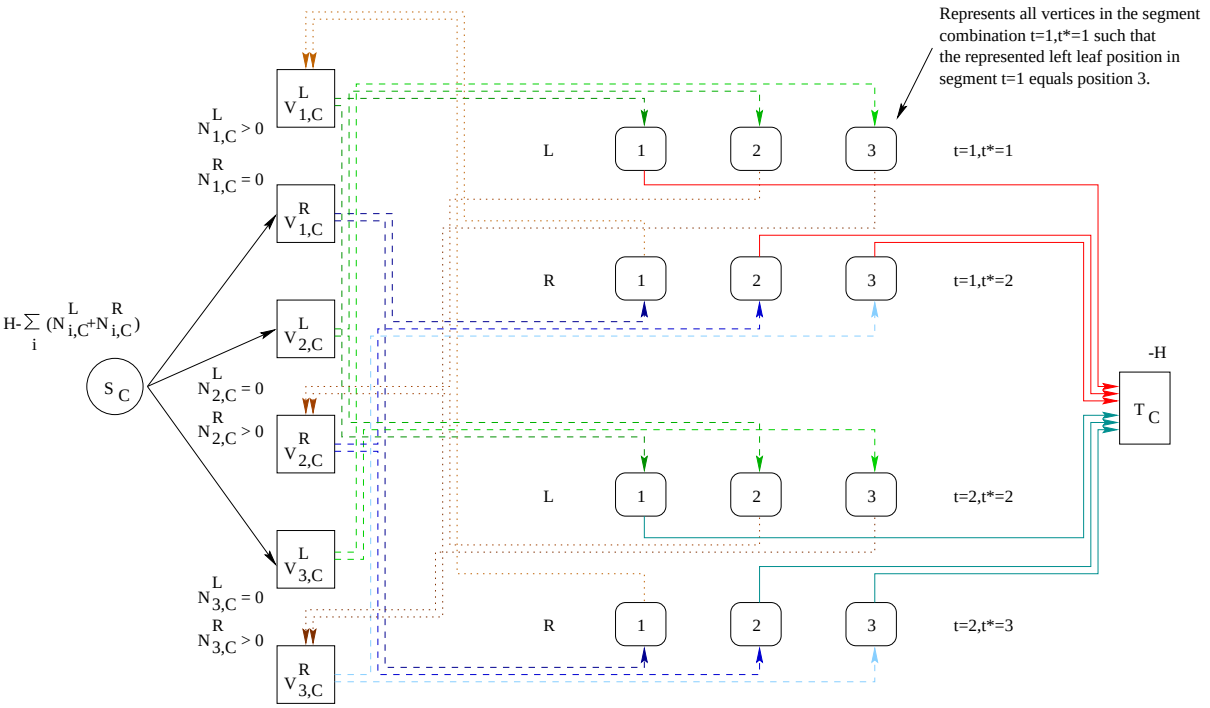


Figure 4.9: *Network for the realization of the intensity profile for an MLC with* $m = 1$, $n = 3$ *and let* $H = 2$.

the number of segments required by the net number of L-leaf positions at i. This is represented by the flow which starts in the newly created source S_C and passes through $V^R_{i,C}$ and now any vertex representing a segment with the R-leaf positioned at i. Finally this unit is routed to $V^L_{i,C}$ in order to start all over again directed towards the sink but defining a segment with an L-leaf positioned at i. Thus, a flow starting in S_C will, when looking at all defined segments, result in a flow which contributes no intensity to the intensity map, which was the aim.

The overall objective is thus to find the lexicographical minimum of

$$\left(\begin{array}{c} \displaystyle\sum_{\substack{(o,p,q,r),(i,j,k,l) \in \mathcal{L}: \\ i>o, j<p, k>r, l<q \\ C \in \{1,\dots,m-1\} \\ t,t^* \in \{1,\dots,H\}, t<t^*}} x^{o,p,q,r,C+1,t,t^*,1}_{i,j,k,l,C,t,t^*,2} \\ \displaystyle\sum_{\substack{(o,p,q,r),(i,j,k,l) \in \mathcal{L}_{tng}: \\ C \in \{1,\dots,m-1\} \\ t,t^* \in \{1,\dots,H\}, t<t^*}} x^{o,p,q,r,C+1,t,t^*,1}_{i,j,k,l,C,t,t^*,2} \\ \displaystyle\sum_{i,j,k,l,C,t,t^*} x^{i,j,k,l,C,t,t^*,2}_{i,j,k,l,C,t,t^*,1} \\ \displaystyle\sum_{\substack{(i,j,i,j) \in \mathcal{L}: i \neq j \\ C \in \{1,\dots,m\} \\ t_p \in \{1,\dots,H\}}} x^{T,t_p,t_p,\downarrow}_{(i,j,i,j),C,t_p,t_p} * C_{D_R} + \displaystyle\sum_{\substack{(i,j,k,l) \in \mathcal{L}: i \neq k \text{ or } j \neq l \\ C \in \{1,\dots,m\} \\ t_p \in \{1,\dots,H\}}} c^{T,t_p,t_{p+1},\downarrow}_{(i,j,i,j),C,t_p,t_{p+1}} * x^{T,t_p,t_{p+1},\downarrow}_{(i,j,i,j),C,t_p,t_{p+1}} * OH_{VR} \end{array} \right)$$

Finally, an optimization formulation of the decomposition problem in terms of a fixed charge network flow problem has been derived, which is known to be $\mathcal{NP}$-hard [Kim & Pardalos (1999)].

In order to conclude that the real world leaf sequencing problem is $\mathcal{NP}$-hard, too, it can be seen that the decision problem to find a decomposition of value less than or equal to an integer k is polynomially transformable to the decision problem asking whether there exists a flow in the previously described network with a value of less than or equal to k. The fact that the transformation can be done in polynomial time can be conluded by the following:
The real world problem's input is an intensity matrix $\mathcal{I} \in \mathbb{Z}^{m \times n}$ and it outputs the placement of the leaves, and thus the shape matrices for each segment which is needed. As the real world problem needs to care for tongue-and-groove effects as well, it can be concluded, that the number of variables required for any model of the real world problem is at least of the order $\mathcal{O}(m * n * N_{Seg})$. The data needed to design the fixed charge problem is of the order $\mathcal{O}$(Number of edges + Number of vertices). This can easily be evaluated. The number of vertices is of the order $\mathcal{O}(m * n^4 * H^2)$, where H is bounded by $\sum_{C=1}^{m} \sum_{j=1}^{n} N_{j,C}^{L}$. Equivalently the number of edges is of the order $\mathcal{O}(m * n^4 * H^2 * n^4)$. Since

$$N_{Seg} \leq m * n * \max_{i,j} \{\mathcal{I}_{i,j}\}$$

and thus

$$H \leq m * n * \max_{i,j} \{\mathcal{I}_{i,j}\}.$$

Additionally, it can be observed, that

$$H \leq m * n * N_{Seg}$$

and thus there exists a polynomial time transformation from the real world problem to the input data of the fixed charge problem.

q.e.d.

Chapter 5

Existing Algorithms

Numerous algorithms have been developed in order to deal with the multileaf collimator problem. A large number of these algorithms addresses the dynamic technique [Convery & Rosenbloom], [Papatheodorou et al.], [van Santvoort & Heijmen], [Svensson et al.]. At this stage, it should be mentioned again, that this thesis was to model the problem as a combinatorial optimization problem, that is why the dynamic part is neglected completely as most of its solution approaches are based on the use of a system of differential equations.
Others, like [Bortfeld et al.(1994)] and [Yu et al.], discuss the optimal position of the leaf in the static decomposition problem when the intensity profile is given as a continuous function. Again, this is an interesting topic to deal with, but the author thinks, that it is by far more likely that the intensity function on the beam heads are given as discrete values on a certain grid. Of course, one could use spline interpolation to approximate a continuous function. But, it has to be taken care of the fact, that the optimization part has to be rerun in this case, as changing the intensity function would definitely cause changes to the quality of the plan, either good or bad.
For the purpose of this thesis, one is mainly interested in those algorithms which output the sequence of leaf positions when dealing with the static case, as it was done in [Galvin et al.(1993)], [Siochi (1999)], [Webb (1998b)] and [Xia & Verhey (1998)].

In the following, the focus is on the algorithms given in [Siochi (1999)] and [Xia & Verhey (1998)]. A detailed description of their algorithms will be given. We will not consider Webb's algorithms as his approach is not robust, that means his routines fail to produce results on certain intensity matrices.

5.1 Siochi's Algorithm

Siochi is trying to find a decomposition of the intensity map $\mathcal{I}$ into shape matrixes $S_i \in \{0,1\}^{m\times n}$, i.e. $\mathcal{I} = \sum_i \alpha_i S_i$, where $\alpha_i \in \mathbb{Z}$ are the relative beam-on time coefficients, which are directly proportional to the number of monitor units which have to be delivered. The number of monitor units (M_i) which have to be delivered for shape matrix S_i can be computed by the following relation

$$M_i = \frac{D}{L}\frac{\alpha_i}{OF_i}$$

where D is the number of monitor units in order to meet the peak fluence of the intensity map ($L = \max_{i,j}\{\mathcal{I}_{i,j}\}$). OF_i is a physical parameter which is depending on the shape of the collimator. Simply speaking a fully opened MLC allows a higher doserate than an MLC with only a single channel opened slightly. This factor will be ignored for simplification. Thus, shape matrix S_i will receive

$$M_i = \frac{D}{L}\alpha_i$$

number of monitor units. Each monitor unit is delivered in constant time, based on the doserate of the linear accelerator. The time needed, in order to deliver a single monitor unit is thus $\dot{D}^{-1}$, where $\dot{D}$ is the doserate of the linear accelerator, usually given in $\left[\frac{\text{monitor units}}{\text{min}}\right]$. Siochi then computes the delivery time (t) as follows:

$$t = \sum_i \frac{M_i}{\dot{D}} + \sum_i \max\left\{OH_{VR}, \frac{\max_{j\in\{1,\ldots,2*m\}}\left\{|x_i^j - x_{i-1}^j|\right\}}{\nu}\right\}$$

where ν is the leaf speed, and x_i^j is the position of the j-th leaf in the i-th segment, and hence $\frac{|x_i^j - x_{i-1}^j|}{\nu}$ is the time the j-th leaf needs to move from its position in segment $i-1$ to its position in segment i.

The algorithm mainly consists of two parts, the first is the *extraction*, and the second is the *rod pushing*. As the extraction routine can only be understood after having explained the rod pushing procedure, the latter will be explained first.

- In the rod pushing procedure Siochi introduces a technique similar to the sweep technique, which was defined by [Bortfeld et al.(1994)], i.e. the leaves sweep on creation of the intensity profile from one side to the other of the multileaf collimator. This drastically reduces the possible number of choices on how to create the given profile. In a first approach, each row of the intensity matrix is processed separately, in order to ensure the delivery of the intensity requirement. As a consequence neither interleaf motion, nor tongue-and-groove corrections can be handled at this stage. Each entry of the intensity map is assigned to a rod, the length of which represents the value in the corresponding entry of the intensity map. By the definition of the sweep technique, it is clear, that after cell j in channel i of the MLC has been opened for irradiation in segment k it will remain open for all subsequent segments until the full intensity requirement of this cell has been delivered.
  Thus, the definition of shape matrices can be reduced to defining a certain base height $b_{i,j}$ for the rods which corresponds to that segment's number, when the MLC will be opened at the corresponding location, and the top of the rod $t_{i,j} = b_{i,j} + \mathcal{I}_{i,j} - 1$ which equals to the number of the last segment with the MLC open in channel i and cell j. The base of the first cell in each channel i will be defined as 1 and thus the top is defined by the height of the first rod, that is $\mathcal{I}_{i,1}$. All following rods in channel i, will be defined by

  $$\begin{aligned} b_{i,j+1} &= b_{i,j} \quad \text{if} \quad \mathcal{I}_{i,j+1} \geq \mathcal{I}_{i,j} \\ t_{i,j+1} &= t_{i,j} \quad \text{if} \quad \mathcal{I}_{i,j+1} < \mathcal{I}_{i,j} \end{aligned}$$

  It is at hand, that after this initial definition of segments, several undesired effects occur, as e.g. interleaf motion and tongue-and-groove effects.

In order to cope with interleaf-motion, the following extension to the basic rod pushing is proposed by Siochi. For each column j determine the rod whose top $t_{i,j}$ is greater than or equal to the tops of all other rods of this column. Starting from the row $i-1$ with decreasing row index, apply the following corrections:

$$\left.\begin{aligned} t_{i+1,j} &= b_{i,j} - 1 \\ b_{i+1,j} &= t_{i+1,j} - \mathcal{I}_{i+1,j} + 1 \end{aligned}\right\} \quad \text{if } b_{i,j} > t_{i+1,j} + 1$$

$$\left.\begin{aligned} t_{i,j} &= b_{i+1,j} - 1 \\ b_{i,j} &= t_{i,j} - \mathcal{I}_{i,j} + 1 \end{aligned}\right\} \quad \text{if } t_{i,j} + 1 < b_{i+1,j}$$

Then, starting from the row just above the "maximum top" row, iterate through all rows with increasing row indices and apply the following corrections, which are similar to the one above:

$$\left.\begin{aligned} t_{i-1,j} &= b_{i,j} - 1 \\ b_{i-1,j} &= t_{i-1,j} - \mathcal{I}_{i-1,j} + 1 \end{aligned}\right\} \quad \text{if } b_{i,j} > t_{i-1,j} + 1$$

$$\left.\begin{aligned} t_{i,j} &= b_{i-1,j} - 1 \\ b_{i,j} &= t_{i,j} - \mathcal{I}_{i,j} + 1 \end{aligned}\right\} \quad \text{if } t_{i,j} + 1 < b_{i-1,j}$$

This procedure is repeated until no further corrections are necessary. After a single column has finished being processed, the sweep technique has to be applied in order to avoid contiguity problems in the columns with higher column indices. Problems may occur when the intensity map includes a "0" entry. This difficulty can easily be overcome by defining the top of this rod to be one unit below its base.

A detailed look at interleaf motion violating setups shows that interleaf motion implies the base of a rod $b_{i,j}$ to be greater than the top of the neighbouring rod $t_{i+1,j}+1$. Since the previously mentioned extension clearly removes all these setups, no interleaf motion can occur.

As discovered previously, the tongue-and-groove effect occurs when adjacent leaf pairs alternately expose adjacent cells in a column. The corrections consist of matching adjacent rods within a column of the intensity solid so that the shorter rod of each pair has its base above or at the same level as the base of the longer rod and has its top below or at the same level as the top of the longer rod. By applying this rule to zero height rods as well , the interleaf motion constraints are simultaneously met as it is impossible that the top of a rod is more than one unit below the base of an adjacent rod in the same column. The corrections are according to the following equation, whereas the process of correction is as the process of corrections for the interleaf motion.

$$\left.\begin{aligned} t_{i+1,j} &= t_{i,j} \\ b_{i+1,j} &= t_{i+1,j} - \mathcal{I}_{i+1,j} + 1 \end{aligned}\right\} \quad \text{if } \mathcal{I}_{i,j} < \mathcal{I}_{i+1,j} \text{ and } t_{i,j} > t_{i+1,j}$$

$$\left.\begin{aligned} b_{i,j} &= b_{i+1,j} \\ t_{i,j} &= b_{i,j} + \mathcal{I}_{i,j} - 1 \end{aligned}\right\} \quad \text{if } \mathcal{I}_{i,j} < \mathcal{I}_{i+1,j} \text{ and } b_{i,j} < b_{i+1,j}$$

$$\left.\begin{array}{ll} t_{i,j} & = t_{i+1,j} \\ b_{i,j} & = t_{i,j} - \mathcal{I}_{i,j} + 1 \end{array}\right\} \quad \text{if } \mathcal{I}_{i,j} \geq \mathcal{I}_{i+1,j} \text{ and } t_{i,j} < t_{i+1,j}$$

$$\left.\begin{array}{ll} b_{i+1,j} & = b_{i,j} \\ t_{i+1,j} & = b_{i+1,j} + \mathcal{I}_{i+1,j} - 1 \end{array}\right\} \quad \text{if } \mathcal{I}_{i,j} \geq \mathcal{I}_{i+1,j} \text{ and } b_{i,j} > b_{i+1,j}$$

- The extraction procedure tries to build shape matrices S_e by the following relation:

$$S_e[i,j] = \begin{cases} \alpha_e & \text{if } \mathcal{I}_{i,j} \geq \alpha_e \\ 0 & \text{if } \mathcal{I}_{i,j} < \alpha_e \end{cases}$$

  However this equation only represents the first step. It is obviously possible that the extracted matrix violates contiguity, interleaf motion and tongue-and-groove constraints. The resulting matrix is analyzed columnwise to find a contiguous set of nonzero entries in a row that either intersects the column or touches the column's right hand side. The same analysis can be done for the column's left hand side. For each column, there is a left shape and a right shape matrix consisting of a single area of consecutive nonzero entries per row, that meets the intersection or touch criteria. The largest of the matrices among all the columns is then used as the extract. By using only those areas with the consecutive ones property that intersect or touch a column on only one side one avoids interleaf motion.
  Again, the process of tongue-and-groove elimination is set up in order to avoid all situations which might possibly lead to a tongue-and-groove effect. Suppose the extracted shape matrix is E and the remaining intensity map is $\mathcal{I}^E$, which is defined by $\mathcal{I}^E := \mathcal{I} - E$. Then no tongue-and-groove effect can occur, if the following two equations are met for all rows r and columns c:

$$\mathcal{I}^E_{r,c} \leq \mathcal{I}^E_{r+1,c}, \text{ if } E_{r,c} = 0 \text{ and } E_{r+1,c} > 0$$
$$\mathcal{I}^E_{r,c} \leq \mathcal{I}^E_{r-1,c}, \text{ if } E_{r,c} = 0 \text{ and } E_{r-1,c} > 0$$

  These equations have to be checked iteratively for all entries of the extracted matrix. Whenever one is violated, then the corresponding entry in the intensity matrix has to be set to zero. As soon as a change has been made to the extracted matrix, the process has to be restarted. After finishing this iteration, one needs to check for the deliverability (i.e. check for physical pairings and interleaf motions). As this might cause changes to the extracted matrix again, one needs to rerun the tongue-and-groove procedure until no further changes will be necessary, neither due to the tongue-and-groove constraints, nor to the deliverability constraints.

The algorithm's optimization routine can then be described as follows: The routine tries several extraction sequences and evaluates each solution according to its delivery time, i.e. each matrix resulting as the difference from the original intensity map and the extracted matrices is then decomposed by the rod pushing procedure. The best of all is then chosen as the final decomposition.

We should remark that it is very hard to describe Siochi's algorithm analytically due to the mixture of procedures and the mixture of interpretations and algorithmic aspects. We have nevertheless tried to explain the algorithm as detailed as possible.

5.2 Xia and Verhey's Algorithm

Xia and Verhey have proposed an algorithm, which differs significantly from the one presented by Siochi. They have actually proposed two algorithms, but both of them produce similar results with respect to the delivery time of the intensity map, thus here the *reducing level technique* is presented. Xia and Verhey propose to decompose the problem into smaller instances as follows. Given a sequence of pairwise different integers $(\alpha_k)_k$, then Xia and Verhey suggest to find the best sequence of physically deliverable shape matrices S_k^i such that

$$\mathcal{I} = \sum_k \alpha_k \left(\sum_i S_k^i \right).$$

They claim to have found that the best predetermined sequence $(\alpha_k)_k$ is based on a sequence of decreasing powers of 2, where the maximum value is found as follows: define a by

$$a := \log_2 \left(\max_{i,j} \{\mathcal{I}_{i,j}\} \right)$$

and b by

$$b := int(a)$$

i.e. b is the rounded off integer closest to a rather than the truncated integer.
Then the delivery sequence is defined as follows:

$$\left(2^{b-1}, 2^{b-2}, 2^{b-3}, \ldots, 2^1, 2^0\right).$$

Hence, the problem can be formulated as follows.

INPUT: Intensity matrix $\mathcal{I} \in \mathbb{Z}^{m \times n}$ and $b := int\left(\log_2\left(\max_{i,j}\{\mathcal{I}_{i,j}\}\right)\right)$
LET $k := b - 1$
REPEAT
    LET $\mathcal{J}_{i,j} := \begin{cases} 1 & \text{if } \mathcal{I}_{i,j} \geq 2^k \\ 0 & \text{otherwise} \end{cases}$
    LET $\mathcal{I} := \mathcal{I} - 2^k * \mathcal{J}$
    Find a good decomposition of $\mathcal{J}$
    LET $k := k - 1$
UNTIL $\max_{i,j}\{\mathcal{I}_{i,j}\} = 0$

It is obvious, that the crucial point is to find a good decomposition of $\mathcal{J}$ in the outline of the algorithms presented above. Xia and Verhey propose to apply a greedy procedure, i.e. they sequentially extract the largest feasible shape matrix from $\mathcal{J}$ until $\mathcal{J}$ has been completely delivered. This can be formulated as follows:

INPUT: $\mathcal{J} \in \{0,1\}^{m \times n}$
REPEAT
    Extract the largest valid shape matrix S from $\mathcal{J}$
    Let $\mathcal{J} := \mathcal{J} - S$
UNTIL S = 0

Unfortunately, the two authors do not state how this largest feasible shape matrix is found. But, a solution can be given easily: it can be found by a linear programming formulation. The set of constraints is given by the subproblem (P_{CG}) and the objective is replaced by maximizing the number of 1-entries. A drawback arises looking at the tongue-and-groove handling which has almost completely been neglected.

5.3 Classification of Exisiting Algorithms

Whereas Siochi's algorithms cannot be classified with respect to the models that have been developped in previous chapters, we can do this with Xia and Verhey's algorithm[1].
Each extraction of $\mathcal{J}$ from the intensity map $\mathcal{I}$ results in a problem similar to the original problem. Additionally, the new problem of decomposing $\mathcal{J}$ can mathematically be formulated as a set partitioning problem by exploiting the fact that decomposing an intensity map with only two intensity levels, namely 0 and any other value a, can be reduced to decomposing a 0-1-matrix $\mathcal{K}$ – chosen such that $a * \mathcal{K} = \mathcal{J}$ – into valid shape matrices S_k with

$$\sum_k S_k = \mathcal{K}$$

Finally, the intensity profile is met by changing the beam-on time per shape matrix from 1 to a, which finally results in

$$a * \sum_k S_k = a * \mathcal{K} = \mathcal{J}.$$

In order to prove the connection to the set partitioning problem the following theorem can be stated:

Here, it should be mentioned, that the problem to decompose the 0-1-matrix can be represented as the continuous relaxation of the given set partitioning problem. But, it shows, that the solutions of this model tend to use many different shape matrices, which increase the total delivery time enormously. Hence, we force degeneracy in the solution by forcing integrality of the solution vector.

Theorem: $(S_k)_k$ is an optimal decomposition of $\mathcal{K}$ if and only if $(a * S_k)$ is an optimal decomposition of $\mathcal{J}$ - optimal with respect to the delivery time of the profile.

Proof: Let $(T_k)_{k=1,\ldots,f_1}$ be a sequence of distinct shape matrices which define, together with a sequence of integers $(\alpha_k)_{k=1,\ldots,f_1}$, an optimal decomposition of $\mathcal{J}$ and let $(S_k)_{k=1,\ldots,f_2}$ be an optimal decomposition of $\mathcal{K}$.

"$\Longrightarrow$"
Suppose

$$\sum_{k=1,\ldots,f_1} OH_{VR} * k + \frac{\alpha_k}{a} * C_{D_R} < \sum_{k=1,\ldots,f_2} (OH_{VR} * k + 1) * C_{D_R}$$

then $(S_k)_{k=1,\ldots,f_2}$ was not an optimal decomposition of $\mathcal{K}$. This yields a contradiction

[1] The following classification works basically for all algorithms using the reducing level technique.

"$\Longleftarrow$"
Suppose

$$\sum_{k=1,\ldots,f_1} OH_{VR} * k + \frac{\alpha_k}{a} * C_{D_R} > \sum_{k=1,\ldots,f_2} (OH_{VR} * k + 1) * C_{D_R}$$

then $(T_k)_{k=1,\ldots,f_1}$, together with $(\alpha_k)_{k=1,\ldots,f_1}$, was not an optimal decomposition of $\mathcal{J}$. This yields a contradiction

q.e.d.

The set partitioning problem can then be based on the formulation

$$\begin{aligned} \min \quad & \underline{1}^T \alpha \\ \text{s.t.} \quad & \mathcal{U}\alpha = \underline{\mathcal{I}} \\ & \alpha \in \{0,1\}^r \end{aligned}$$

where in this case $\underline{\mathcal{I}}$ is obtained by

$$\underline{\mathcal{I}}_{(i-1)*n+j} = \mathcal{K}_{i,j}.$$

By deleting those constraints i with $\underline{\mathcal{I}}_i = 0$ and elimiating all those variables which correspond to a column j of the original matrix $\mathcal{U}$ where $\mathcal{U}_{ij} = 1$, we gain a real set partitioning problem.

Thus it can be concluded that each "small" decomposition problem which has to be solved in each iteration of Xia and Verhey's algorithm can be formulated as the column generation problem given in a previous chapter and, additionally, the master problem of which is actually a set partitioning problem after reducing the problem size by eliminating redundant and useless constraints and variables [2].

In order to illustrate the hugeness of the given problem we will shortly look at the number of variables and the number of constraints which are included in the set partitioning problem for a clinical example, i.e. with a collimator resolution with $m = n = 10$. It can easily be seen that the number of rows is $n * m$ when no tongue-and-groove constraints are added. This number is considerably small in comparison with the number of variables, which is about 10^{19}, where this is already the reduced amount, i.e. this is the total amount of valid shape matrices. Clearly, the total number of physical pairings without caring for interleaf motion is

$$\left(\frac{(n+1)(n+2)}{2}\right)^m.$$

This formula can be obtained by the following idea: the left leaf in each channel can be placed at $n+1$ positions. Given a position k of the left leaf, the opposing right leaf can be positioned only at those positions which are further to the right than position k. Thus, $n+1-k+1$ positions remain, and so the total number of possible pairings per channel is

$$\sum_{k=1}^{n+1} n+1-k+1 \quad = \quad \sum_{k=1}^{n+1} n+2-k$$

[2] A good overview of methods to tackle set partitioning problems is given by [Balas & Padberg].

$$\begin{aligned} &= (n+1)(n+2) - \sum_{k=1}^{n+1} k \\ &= (n+1)(n+2) - \frac{(n+1)(n+2)}{2} \\ &= \frac{(n+1)(n+2)}{2} \end{aligned}$$

By neglecting interleaf motion, each channel is configured independently and thus the total number of valid shape matrices is

$$\left(\frac{(n+1)(n+2)}{2}\right)^m.$$

Now, it is of importance to subtract all those shape matrices which violate the interleaf motion constraints. Suppose the right leaf in channel 1 is positioned at position k. Then all pairings in channel 2 are forbidden with the left leaf further to the right than position k. This can be derived similarly to the total number of combinations by reducing the channel size and thus the number of forbidden combinations is here $\frac{(n+1-k)(n+1-k+1)}{2}$. As this holds for each position k and for the right and left leaf, we have the following number of forbidden combinations in a 2-channel MLC

$$2 * \sum_{k=1}^{n+1} \frac{(n+1-k)(n+1-k+1)}{2}.$$

Finally, the total number of forbidden combinations in an m-channel MLC is given by

$$\sum_{s=2}^{m} \left(\frac{(n+1)(n+2)}{2}\right)^{m-s} * 2 * \sum_{k=1}^{n+1} \frac{(n+1-k)(n+1-k+1)}{2}$$

and hence the number of feasible shape matrices can be calculated by evaluating

$$f(m,n) := \left(\frac{(n+1)(n+2)}{2}\right)^m - \sum_{s=2}^{m} \left(\frac{(n+1)(n+2)}{2}\right)^{m-s} * \sum_{k=1}^{n+1} (n+1-k)(n+1-k+1).$$

The following table illustrates how fast this function in n and m increases.

m	n	$f(m,n)$
3	3	780
4	4	40985
5	5	$3.403 * 10^6$
6	6	$4.105 * 10^8$
7	7	$6.792 * 10^{10}$
8	8	$1.478 * 10^{13}$
9	9	$4.094 * 10^{15}$
10	10	$1.407 * 10^{19}$

This allows stating that most of the famous algorithms to solve large scale set partitioning problems will fail as they will have to cope with unavoidable computational explosions. This is due to the fact, that most of them require to have the complete coefficient matrix at hand (cf. [Balas & Padberg], [Atamtürk et al.], [Borndörfer]) which is in this application not always possible. It should be stressed again, that $f(m,n)$ represents the number of columns, and thus the number of variables, of the set partitioning problem.

Chapter 6

Application

There are two major factors which motivate the use of a heuristic method rather than solving the problem to optimality. First of all, the fact that the problem has been proven to be $\mathcal{NP}$-hard. Secondly, the huge size of the problem, especially the number of variables, will force this idea, too.

Nevertheless, it was of course tried to obtain an optimal solution of P_{Master} together with its subproblem P_{CG}, and P^{tng}_{Master} together with its subproblem P^{tng}_{CG}. As both problems produced similar results, the observations that were made with P^{tng}_{Master} and its corresponding subproblem P^{tng}_{CG} will be discussed in the following section.

6.1 An Approach To Finding The Optimal Solution

On solving the linear program P^{tng}_{Master} it is needed to generate a column for each pivot iteration using the integer program P^{tng}_{CG}. Although the objective is to minimize the beam-on time, [Siochi (1999)] has claimed that minimizing the relative beam-on time coefficients is a good measure for minimizing the total delivery time. This could be contradicted by numerical runs on test sets. These test runs have shown, that the closer one gets to the optimal solution of the linear program P^{tng}_{CG} the less degenerate the solution is, i.e. more and more segments were used to realize a short beam-on time. In almost all test runs, almost all of the $m * n$ possible segments were used. Calculating the total delivery time, by adding the setup time to the optimal objective function value made it possible to compare the results of this approach to those which were obtained by existing heuristics, especially the ones proposed in [Siochi (1999)] and [Xia & Verhey (1998)]. The comparison shows, that our beam-on time was minimal among all. This is not surprising, as this was the objective to be minimized. But, due to the fact that many more segments were used in this approach the total delivery time was much higher than those produced by the decomposition heuristics.

It is quite interesting to see that other authors have observed this fact as well for large scale set partitioning problems. [Anbil et al.] have observed that small set partitioning problems were massively degenerate but large scale ones were not. Additionally, he observed that the continuous relaxations of large scale set partitioning problems are not very likely to have integer solutions as relaxations of small set partitioning problems are, which is contradicting a folklore saying that linear programming solutions of large scale set partitioning problems are usually integer or close to integer. We have observed a similar behaviour. Our problem is thought to be

highly degenerate but in fact it is not and none of the solutions had a variable that was integer or close to an integer. That is the main reason, why the idea of solving the whole problem in one step has been postponed in its development.

In order to overcome this drawback, it was found, that the presented approach would in general work well, if the setup time had been modelled. This is generally possible, as shown in a previous chapter, but it turns the master problem into a mixed integer programming problem. Observing, that the time required to minimize the beam-on time for a 10×10 MLC leaf sequencing problem with 15 intensity levels took about 3 hours using AMPL[1] and CPLEX 4.0[2], it was considered to be computationally very expensive to solve the master problem with the integral property. Hence, it was our aim to apply our mathematical model to the existing heuristics, in such a way that better solutions with respect to the total delivery time are produced.

6.2 Improving

We will concentrate on improving the algorithm based on the reducing level technique as described by [Xia & Verhey (1998)]. The main idea is to model each iteration in their algorithm as a set partitioning problem and to approach the solution with methods that have been developed to solve large scale set partitioning problems. Here, it should be stressed, that the original column generation problem defined in the very beginning, did not require each α_t to be an integer. The reason, why it is now forced to be binary is to force the solution to be more degenerate in order to overcome the problems that were faced in the previous section - it turned out to be a good decision. The set partitioning is in fact an integer programming problem of a size not much smaller than the original problem, but its special properties convinced us of applying methods that produce good, rather than optimal solutions in a fairly satisfying time.

6.2.1 Applying Gilmore and Gomory's Cutting Stock Method

Gilmore and Gomory have developed a heuristic to obtain a good solution of the cutting stock problem - a huge integer programming problem [Lasdon]. This procedure can be explained using a set partitioning problem as well. Suppose the set partitioning problem is given by

$$\begin{array}{lll} & \min & c^T x \\ (SPP) & \text{s.t.} & Ax = \underline{1} \\ & & x \in \{0,1\}^k \end{array}$$

with an exponential number of variables (k) and $(A)_{i,j} \in \{0,1\}$. It is now attempted to solve the continuous relaxation of (SPP), namely

$$\begin{array}{lll} & \min & c^T x \\ \overline{(SPP)} & \text{s.t.} & Ax = \underline{1} \\ & & x \in [0,1]^k \end{array}$$

[1] AMPL was developed by the Bell Laboratories, a Division of Lucent Technologies, Murray Hill, NJ, U.S.A

[2] CPLEX is a registered trademark of ILOG, Gentilly Cedex, France, and Mountainview, CA, U.S.A.

This is done using a special column generation technique which avoids too many calls of the subproblem. Especially, when the column generation subproblem is hard to solve, then the following algorithm is more efficient.
Suppose that at the current iteration, a subset $J \subset \{1, \ldots, k\}$ of columns of A has already been generated and that the restricted relaxation of $\overline{(SPP)}$

$$\begin{array}{lll} & \min & c_J^T x \\ \overline{(SPP)}_J & \text{s.t.} & A_J x_J = \underline{1} \\ & & x_J \in [0,1]^{|J|} \end{array}$$

is solved exactly by the simplex algorithm. Note that c_J and A_J denote the cost vector c and the constraint matrix A restricted to the subset of variables J, respectively. Solving this relaxation to optimality, yielding x_J^{opt}, is not sufficient to claim that x_J^{opt} is an optimal solution of $\overline{(SPP)}$, and hence an optimality test has to applied. The next step is thus to solve the column generation subproblem, whose aim is to find the column v, which is the k-th column of A, with minimal reduced cost among all columns of A. Suppose the optimal solution of this subproblem has an objective value not less than zero. Then no column can be found which could possibly decrease the objective of the master problem and hence optimaliy of x_J^{opt} to $\overline{(SPP)}$ is proven. In contrast to this, when the objective value of the optimal solution of the subproblem is less then zero, then this column v will be added to the restricted constraint matrix A_J as the $|J|+1$-st column and x_k will be added to the restricted set of variables x_J, again as the $|J|+1$-st variable. Then, the new restricted relaxation $\overline{(SPP)}_J$ is solved again. These steps are repeated until optimality of $\overline{(SPP)}$ is attained. Thus, the optimality test and the column generation procedure which is used to proceed in the optimization process are done in one step.
Finally, integrality is restored and $\overline{(SPP)}_J$ is solved using only binary variables.

One drawback of using this heuristic is that solving the latter with binary variables might break down as no feasible solution can be found. By using a basic feasible starting solution of (SPP) for the initial definition of J one can overcome this drawback, though the quality of the integer solution might still be relatively bad. But, this technique has been successfully applied by [Ribeiro et al.]. In fact, this procedure is likely to find an optimal solution of $\overline{(SPP)}$ on generating less columns – by making use of all columns that have been generated so far – than in the original column generation procedure where each generated column is pivoted into the basis and the leaving columns are deleted. The main reason for applying this technique is the fact, that in the end an integer solution has to be attained. In the usual column generation process it is not at hand how a good integer solution can be found after having solved the continuous relaxation. The given approach is preferred because an integer solution can easily be obtained at the very end of the procedure.

This method can easily be adapted to be applied to our problem formulation P_{Master} and P_{CG} together with a right hand side input of the reducing level technique.

6.2.2 Greedy Heuristics in The Column Generation Process

Depending on the size of the set partitioning problem and on the number of generated columns in the algorithms described above, one needs to adjust things a bit, as the time required to solve

the integer programming problem at the end might still be very unsatisfactory. On the other hand, it is possible, that most of the columns used in the optimal solution of the continuous relaxation were useless for a good integer solution. [Barnhart, Johnson et al.] and [Anbil et al.] have proposed an algorithm which completely avoids the exact solution of any integer program by using a heuristical branching scheme in order to obtain an integer solution at the end. This is done as follows.

They propose to solve the first instance of the continuous relaxation based on a given set of columns as described above. Then, they fix those variables with a high fractional part permanently to 1 and resize, i.e. reduce the problem dimension, and rerun the optimization on the new problem again until all unfixed variables must have a value of zero as the right hand side of the partitioning problem is already covered by the fixed variables. Yet, both authors report differently on the application of this search strategy. [Anbil et al.] claims that this strategy fails on large crew pairing problems, whereas [Barnhart, Johnson et al.] claim, that it has been successfully applied by [Marsten].

When fixing variables to 1, one needs to care for maintaining feasibility. It is not explicitely stated how this is done, but one possibility can easily be derived. Obviously, fixing all those variables to 1, whose fractional part is strictly larger than 0.5 will not cause any trouble. But, there might not necessarily be such a variable. In this case, we suggest to fix the variable with the largest fractional part to 1. If there are several, then it should be the one corresponding to the column with the largest number of 1-entries.

As in the latter case one was greedy on the fractional part of the variables, we will now look at a different criterion that selects the variables which are fixed to one.

In order to decrease the problem size as much as possible by each variable fixing, one can be greedy simply by looking at the column with the largest number of 1-entries among all columns with a positive value in the optimal solution of the continuous relaxation, which defines a second greedy procedure.

In [Barnhart, Johnson et al.] another approach has been discussed, which is similar to the ones already presented. The two greedy approaches and the following one are all based on a column generation technique in which the branch-and-bound tree is greedily searched according to a given criterion. As the new approach is no longer fixing variables to one, we will introduce a new section.

6.2.3 A Heuristic Based on A Proposal by Barnhart, Johnson et al.

[Barnhart, Johnson et al.] state a proposition which defines the branching constraints for set partitioning problems.

Proposition: If Y is a 0-1 matrix, and a basic solution to $Y\lambda = 1$ is fractional, i.e., at least one of the components of λ is fractional, then there exist two rows r and s of the master problem

such that

$$0 < \sum_{k:Y_{rk}=1,Y_{sk}=1} \lambda_k < 1.$$

The straight forward proof can be found in [Barnhart, Johnson et al.].

The pair r, s gives the pair of branching constraints

$$\sum_{k:Y_{rk}=1,Y_{sk}=1} \lambda_k = 1 \text{ and } \sum_{k:Y_{rk}=1,Y_{sk}=1} \lambda_k = 0$$

i.e. the rows r and s have to be covered by the same column on the first branch and by different columns on the second branch. As long as a branching pair can be found in the set of rows, the solution must be fractional. Since there is only a finite number of pairs of rows, the branch-and-bound algorithm terminates after a finite number of iterations.

This branch-and-bound algorithm can easily be translated into a heuristic to improve the current basic feasible solution. For the applications of set partitioning problems, it is of major importance to find a good feasible solution rather than proving optimiality. Therefore, [Barnhart, Johnson et al.] claim that it usually makes sense to choose a branching decision that devides the solution space in such a way, that one is more likely to find a good solution in one of the two nodes created and then choose this node for evaluation first. That means, one greedily searches the branch-and-bound tree on the branch that is more likely to yield a good feasible integer solution. For set partitioning problems, it is therefore essential, to choose the rows r and s such that

$$\sum_{k:Y_{rk}=1,Y_{sk}=1} \lambda_k$$

has a value which is close to one. This yields a primal heuristic by ignoring all other branches.

On implementing this approach difficulties to handle our huge scale problems were faced. Small problems could be solved and the solutions that were obtained were very convincing. But, due to the fact, that after each addition of a new branching constraint a lot of columns had to be added to the original model in order to maintain feasibility, the size of the problems to solve the continuous relaxation and the size of the subproblem to determine a profitable column increased. The latter increased due to the new constraints that had been added. Testing a decomposition problem on a 10×10 intensity matrix not a single set partitioning problem stemming from the reducing level technique could be solved within 6 hours of CPU time. Thus, this heuristic was neglected in the chapter of numerical results. This drawback is not as bad as it looks, looking at the quality of the solutions obtained by our first greedy approach. As this heuristic is based on the reducing level technique as well, one can observe that a significant improvement cannot be made as the decomposition results from the first greedy approach are very close to their theoretical optimum. This will be displayed in the next section.

Chapter 7

Numerical Results

This section addresses the numerical results that were obtained on the application of the algorithms and gives a detailed comparison and overview. In order to compare the different results of the various algorithms all algorithms were run without the option of reducing the tongue-and-groove effect. We will display the results of the algorithms individually and afterwards compare them. All algorithms were tested on 15 randomly generated 10×10 intensity matrices with 15 intensity levels, reaching from 1 to 15. The paper, which describes reality the most detailed is, from our point of view [Siochi (1999)]. Hence, we will set the necessary parameters as stated there. Most of these parameters are only necessary in order to compute the total delivery time after the algorithms has output the decomposition of the intensity profile. Though, Siochi's algorithm does actually output different sequences, when the parameters are changed. The following list, describes the parameters which have to be set and which value we will assign to them.

- The number of monitor units per intensity level is set to 2.
- The linear accelerator tis able to deliver 200 monitor units per minute, i.e. delivering a single monitor unit takes $\frac{60}{200}$ sec.
- Hence $C_{D_R} = 2 * \frac{60}{200} = 0.6$
- The V&R-overhead, i.e. the setup time between two non-identical segments will be set to 18 sec and later on to 4 sec.

7.1 Results Obtained by Siochi's Algorithm

For the theoretical base of this algorithms we refer to the chapter where the existing heuristics have been described and to [Siochi (1999)]. The following table will state the results, that were obtained with a V&R-overhead of 4 sec.

Test set	The number of different shape matrices	Sum of the relative beam-on time coefficients	Resulting Delivery Time in [sec]
1	30	50	150.0
2	27	40	132.0
3	27	42	133.2
4	27	39	131.4
5	28	58	146.8
6	28	51	142.6
7	28	52	143.2
8	29	48	144.8
9	24	40	120.0
10	26	47	132.2
11	29	47	144.2
12	29	43	141.8
13	22	40	112.0
14	27	47	136.2
15	27	42	133.2

Using the same test sets, but changing the V&R-overhead from 4 to 18 sec, then the following results are obtained

Test set	The number of different shape matrices	Sum of the relative beam-on time coefficients	Resulting Delivery Time in [sec]
1	30	50	570.0
2	27	40	510.0
3	27	42	511.2
4	27	39	509.4
5	28	58	538.8
6	28	51	534.6
7	28	52	535.2
8	29	48	550.8
9	24	40	456.0
10	26	47	496.2
11	28	60	540.0
12	29	43	547.8
13	22	40	420.0
14	27	47	514.2
15	27	42	511.2

We observed, that in almost all test sets the decomposition was identical, regardless of the change in the overhead time. Though, we observed that, when the doserate was decreased significantly than the decomposition features more segments and the beam-on time decreased. These results were obtained using a C++ code on a PC with a 133 MHz Intel Pentium[1] processor. The computational time to obtain any of these results were in no case larger than 2 minutes.

All remaining algorithms have been implemented using AMPL[2] and CPLEX 4.0[3] on an IBM RS/6000[4], Model G 30 (two-way), as each depends heavily on the solution of linear programs.

Additionally, all remaining algorithms don't allow the input of the specifications of the linear accelerator and hence the intensity profile is the only input. The decomposition obtained is then evaluated according to the parameters stated above.

[1] Pentium is a registered trademark of the Intel Corporation, Santa Clara, CA, U.S.A.

[2] AMPL was developed by the Bell Laboratories, a Division of Lucent Technologies, Murray Hill, NJ, U.S.A

[3] CPLEX is a registered trademark of ILOG, Gentilly Cedex, France, and Mountainview, CA, U.S.A.

[4] IBM and RS/6000 are registered trademarks of the International Business Machines Corporation, Armonk, NY, U.S.A.

7.2 Results Obtained by Xia and Verhey's Algorithm

For the theoretical base of this algorithms we refer to the chapter where the existing heuristics have been described and to [Xia & Verhey (1998)]. The table below states the results, that were obtained by decomposing the 15 different test sets.

Test set	The number of different shape matrices	Sum of the the relative beam-on time coefficients	Resulting Delivery Time in [sec] with $OH_{VR} = 4$ sec	Resulting Delivery Time in [sec] with $OH_{VR} = 18$ sec
1	22	87	140.2	448.2
2	21	79	131.4	425.4
3	23	95	149.0	471.0
4	24	87	148.2	484.2
5	22	81	136.6	444.6
6	21	77	130.2	404.4
7	22	87	140.2	424.2
8	21	73	127.8	484.2
9	22	87	140.2	421.8
10	23	89	145.5	448.2
11	20	71	122.6	467.4
12	26	93	159.8	402.6
13	19	69	117.4	523.8
14	23	93	147.8	383.4
15	24	93	151.8	469.8

The time to obtain any of the decompositions did not exceed 5 min.

7.3 Results Obtained by The Application of The Cutting Stock Procedure

Here, we will show the numercial results, that have been obtained by the application of the procedure to obtain a good integer solution using the method of Gilmore and Gomory to solve the cutting stock problem. In order to compare the results, the input of the procedure are the same 15 randomly generated matrices as for the other strategies.

Test set	The number of different shape matrices	Sum of the the relative beam-on time coefficients	Resulting Delivery Time in [sec] with $OH_{VR} = 4$ sec	Resulting Delivery Time in [sec] with $OH_{VR} = 18$ sec
1	45	188	292.8	922.8
2	44	131	254.6	870.6
3	38	126	227.6	759.6
4	46	174	288.4	932.4
5	43	170	274.0	876.0
6	44	159	271.4	887.4
7	45	171	282.6	912.6
8	43	158	266.8	868.8
9	44	172	279.2	895.2
10	44	147	264.2	880.2
11	41	124	238.4	812.4
12	50	198	318.8	1018.8
13	35	134	220.4	710.4
14	49	171	298.6	984.6
15	35	116	209.6	699.6

The computational time to decompose a single intensity matrix was in average about 2 hours. As this application of the cutting stock method is based on the reducing level technique, it could be parallelized into l procedures, where l is the number of levels used in the reducing level technique. In this case, there were 4 levels, and thus a sophisticated new computer could probably solve the problem within 20 to 30 min. We will create the extracts corresponding to the levels first and then solve each of them in parallel. This is possible as each extract is decomposed such that it is completely delivered and so the extracts do not influence each other.

7.4 Results Obtained by The Greedy Strategies

In the first step we will display the results, that have been obtained using the greedy strategy in which one fixes the variables with a high fractional part to one. Again, the same 15 randomly generated intensity matriced were used as input for this routine and the delivery times for a 4 and 18 sec overhead, respectively, were calculated.

Test set	The number of different shape matrices	Sum of the the relative beam-on time coefficients	Resulting Delivery Time in [sec] with $OH_{VR} = 4$ sec	Resulting Delivery Time in [sec] with $OH_{VR} = 18$ sec
1	23	92	147.2	469.2
2	19	64	114.4	380.4
3	20	75	125.0	405.0
4	19	67	116.2	382.2
5	21	76	129.6	423.6
6	20	74	124.4	404.4
7	20	79	127.4	407.4
8	17	61	104.6	342.6
9	22	80	136.0	444.0
10	21	76	129.6	423.6
11	18	65	111.0	363.0
12	21	81	132.6	426.6
13	19	68	116.8	382.8
14	22	78	134.8	442.8
15	19	77	122.2	388.2

In fact, this strategy performs very good as can be seen by comparing the results to other heuristics. In almost all cases, this strategy performs better than all other algorithms. As a reason for this, we could observe that this greedy strategy to obtain a solution of the set partitioning problem does in many cases reach the optimal of each set partitioning problem. This can be verified as the objective value of the integer solution obtained equals in many cases the optimal objective value of the continuous relaxation. A more detailed analysis will be given later.

On the other hand, this stratey belongs to those being computationally highly expensive, as it took about 3 hours to decompose a single intensity matrix. Parallelizing the process as indicated above could in this case decrease the effort to about 40 to 50 min of computational time.

We will now shortly look at the second greedy approach. As a short review, in this approach we are greedy on the size of the shape matrix when fixing variables to one in the solution process of each set partitioning problem. Beside this small change, everything else is similar to the first greedy heuristic and the results that were obtained are presented in the table below.

Test set	The number of different shape matrices	Sum of the the relative beam-on time coefficients	Resulting Delivery Time in [sec] with $OH_{VR} = 4$ sec	Resulting Delivery Time in [sec] with $OH_{VR} = 18$ sec
1	25	106	163.6	513.6
2	22	77	134.2	442.2
3	25	107	164.2	514.2
4	26	86	155.6	519.6
5	22	88	140.8	448.8
6	24	98	154.8	490.8
7	23	85	143.0	465.0
8	27	104	170.4	548.4
9	25	112	167.2	517.2
10	27	101	168.6	546.6
11	26	101	164.6	528.6
12	29	114	184.4	590.4
13	20	78	126.8	406.8
14	25	95	157.0	507.0
15	21	73	127.8	421.8

Again, the computational time is very large, in average slightly larger than the first greedy approach. It took about 3.5 hours in average for each decomposition.

7.5 Comparison

From the results obtained on the basis of 15 randomly generated 10×10 intensity matrices with 15 intensity levels, it can be observed, that the first greedy strategy, which has been proposed, produces in many cases the best results among all algorithms. Therefore we will compare its results to those obtained by applying Siochi's and Xia and Verhey's algorithms. The following diagrams summarize the delivery times for the three algorithms for each of the 15 intensity matrices, which clearly outlines the goodness of our approach.

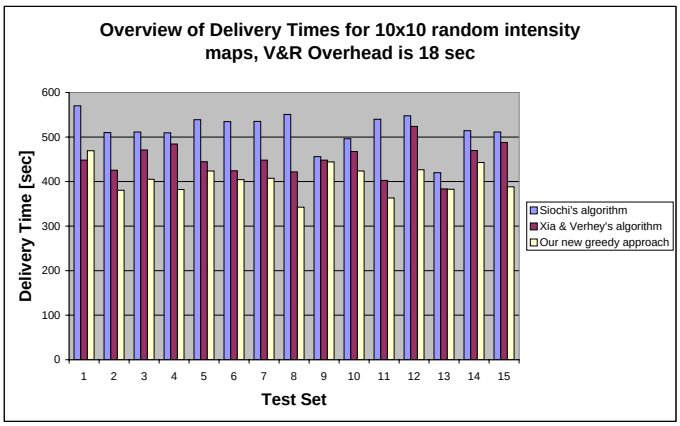


Figure 7.1: *Overview of Delivery Times with a V&R-Overhead of 18 sec*

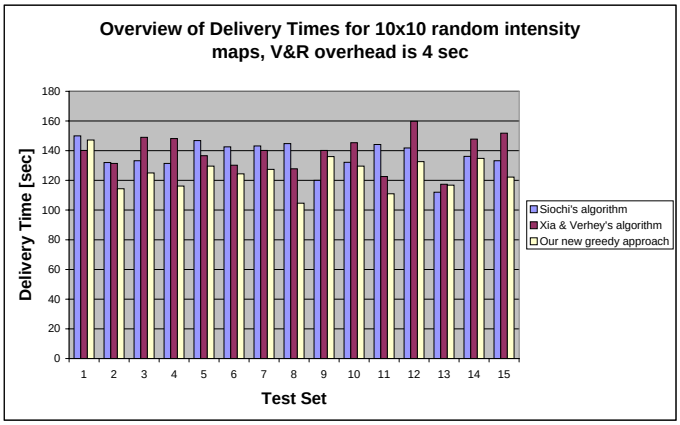


Figure 7.2: *Overview of Delivery Times with a V&R-Overhead of 4 sec*

Looking statistically at the result, our proposed greedy approach needed in average less segments and all in all less time to deliver the intensity matrix than the other algorithms. Due to the fact, that in average the beam-on time contributes about or less than 10% of the delivery time,

this yields the fast delivery of the profile using the greedy approach. A short overview of the average number of different shape matrices, the average beam-on time and the average delivery time together with the standard deviation from the average, gives more insight into the results.

	OH_{VR}	Siochi's Algorithm	Xia and Verhey's Algorithm	Our greedy Approach
Average number of different shape matrices	4 sec	27.2	22.2	20.0
	18 sec	27.1	22.2	20.0
Average beam-on time [sec]	4 sec	45.7	84.0	74.2
	18 sec	46.6	84.0	74.2
Average delivery time [sec]	4 sec	136.2	139.2	123.8
	18 sec	516.4	450.0	405.7
Std. Dev of the number of different shape matrices	4 sec	2.0	1.7	1.6
	18 sec	2.0	1.7	1.6
Std. Dev. of the beam-on time	4 sec	5.5	8.5	8.0
	18 sec	6.6	8.5	8.0
Std. Dev. of the delivery time	4 sec	10.3	11.7	11.0
	18 sec	38.1	35.8	33.6

The two following boxplots help to summarize this data and give a good overview. The red box indicates the range in which 50 % of all datapoints lie. The black bar in the red box gives the median of all datapoints, that is in our case the median of the delivery times. The upper and lower bar connected to each red box indicate the highest and the lowest delivery time.

7.6 Ratio Analysis of The Greedy Approach

In fact it is so far an open question how good the suggested greedy approach performs in comparison to the global optimum. Here it should be stressed again, that we have of course no guarantee for reaching the global optimum of the overall decomposition problem due to the splitting into several independent smaller set partitioning problems. On the other hand, one is keen to know how good this greedy approach approximates the optimal solution of each set partitioning problem. Therefore one can look at the value obtained by solving the continuous relaxations of the set partitioning problems which gives a lower bound on the binary solution of the latter. Hence, solving the continuous relaxation of the set partitioning problems that were encountered by the 15 test sets used earlier, we can outline the goodness of the approach in the following table:

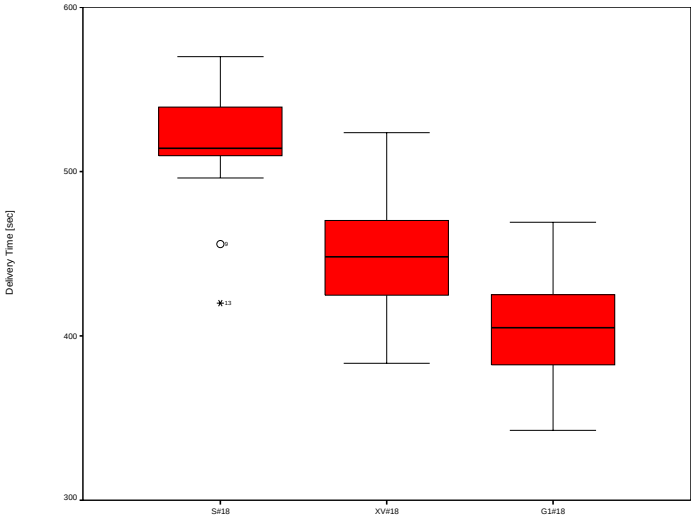


Figure 7.3: *Boxplot of delivery times with a V&R-overhead of 18 sec. The labels are marked as follows: S#18 (Siochi's algorithm with* $OH_{VR} = 18$*), XV#18 (Xia and Verhey's algorithm with* $OH_{VR} = 18$ *and G1#18 (our first greedy approach with* $OH_{VR} = 18$*).*

Test set	The number of different shape matrices in the cont. rel.	Sum of the the relative beam-on time coefficients in the cont. rel.	Gap in the number of different shape matrices	Gap in the sum of beam-on time coeffcients
1	19	74	17%	20%
2	17	61	11%	5%
3	19	73	5%	3%
4	19	67	0%	0%
5	19	74	10%	3%
6	19	73	5%	1%
7	19	71	5%	10%
8	17	61	0%	0%
9	18	66	18%	18%
10	20	75	5%	1%
11	18	65	0%	0%
12	19	71	10%	12%
13	15	56	21%	18%
14	20	75	9%	4%
15	18	69	5%	10%

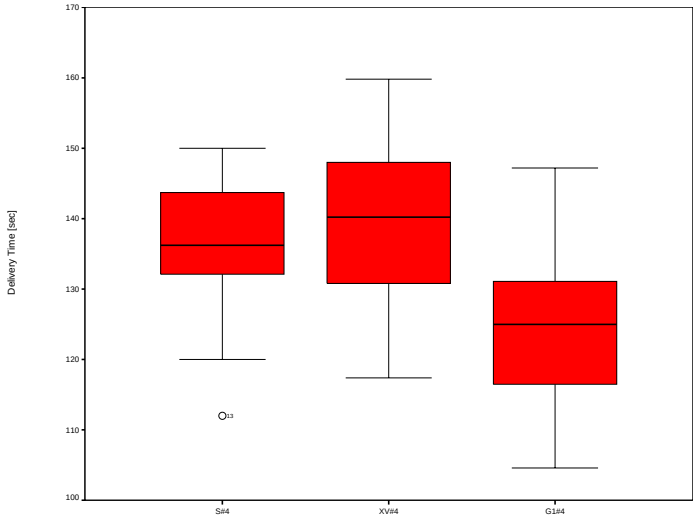


Figure 7.4: *Boxplot of delivery times with a V&R-overhead of 4 sec. The labels are marked as follows: S#4 (Siochi's algorithm with $OH_{VR} = 4$), XV#4 (Xia and Verhey's algorithm with $OH_{VR} = 4$ and G1#4 (our first greedy approach with $OH_{VR} = 4$).*

7.7 Trying to Improve The Performance Using Barahona's Algorithm

As the computational times in the greedy approach were quite high, especially for solving the very first instance of every set partitioning relaxation, we were convinced to use the algorithm proposed by [Barahona& Anbil] in order to gain an optimal primal solution to the continuous relaxation of the set partitioning problem. This algorithm uses a variant of the subgradient algorithm to produce an optimal solution of the problem. Usually, subgradient algorithms do produce an optimal dual rather than primal solution, and in order to compute the values for the primal variables, one has to use a different procedure, e.g. using the complementary slackness conditions. [Barahona& Anbil] have proposed a subgradient algorithm which does in most cases converge [5] to an optimal primal solution. The algorithm works well for some of our set partitioning problems. But we observed that it actually takes longer than our original procedure using CPLEX to solve each relaxation. In some cases we observed that the stepsize was almost equal to zero, although primal feasibility was not reached, and hence no further improvement could be made in the iterative process to create the primal solution. The long CPU time, that we have observed could be caused by AMPL, where we implemented this scheme in order to still be able to handle the column generation subproblem easily, which means the need to obatin a solution of a complicated integer programming problem.

[5] The convergence could not be proven.

Chapter 8

Conclusion and Final Remarks

8.1 Summary

We have developed mathematical models for the static multileaf collimator leaf sequencing problem and we have proven the problem to be $\mathcal{NP}$-hard. Furthermore we have proposed a column generation solution approach that improves the quality of the solutions obtained by the reducing level technique. Although computational times were not satisfactory in comparison to the performance of other algorithms, we are most confident to say that parallelizing the reducing level technique leads to significant improvements in its performance.

8.2 Open Questions

A lot of questions arose during the last months. In order to look ahead and be aware of certain problems and to start discussions about certain aspects of this thesis a list of open questions is provided. Any contribution to these questions is welcome and everyone who thinks about contributing is highly encouraged to share his or her ideas with us.

Concerning the mathematical model

- We think that the partitioning approach is a good approach. Though, we do not claim to have found the best. Therefore it is necessary to verify whether a better approach can be found. In this case, does a new model yield algorithms which produce solutions that are as good or even better than those presented, but perform faster.
- A next step would be to contact MLC manufacturers and discuss the importance of certain constraints with respect to future developments, which might simplify the model enormously.
- It can of course be questioned whether the objective function which is proposed by [Siochi (1999)] is the best for all cases. Certain clinical cases might need the minimization of the beam-on time instead.

Concerning the overall decomposition problem

- An attempt to find a good solution of the overall decomposition problem without decomposing it using the reducing level technique has been done but problems were faced in

the application, since the predicted degeneracy was not present as we were encouraged to hope. It is therefore necessary to think about a different global approach.

- It is not yet known how much of the overall optimality is lost even if each stage of the reducing level technique is solved to optimality. This would be an interesting topic to discuss.

- The original problem was decomposed into several smaller problems, which were not connected to each other and thus tongue-and-groove effects could unfortunately not be considered. It will be necessary to develop a global heuristic approach which enables the elimination of the tongue-and-groove effect. Here it should be stressed that a discussion with Ms Ping Xia from the University of California at San Francisco at the 13th International Conference on the Use of Computers in Radiation Therapy in Heidelberg stressed that the definition of the tongue-and-groove effect in the static case does not seem to be unique.

Concerning the heuristics that have been applied to solve huge scale set partitioning problems

- A worst case analysis of these heuristics has not been undertaken yet. So, one might ask whether there is a worst-case bound of their performance.

- Are their eventually algorithms which perform faster and yield as good or even better approximations of the optimal set partitioning solution?

Concerning the implementation

- All heuristics that were dependent on a linear programming solver were completely implemented in AMPL. As each call of CPLEX[1] in AMPL will cause the translation of the algebraic model into an MPS File, a lot of computational time is wasted. For practical use, the heuristics should be implemented using C/C++ and CPLEX should be called directly, which enables changing the relevant part of the model in the computer's memory, rather than building a completely new model on its disk.

[1]CPLEX is a registered trademark of ILOG, Gentilly Cedex, France, and Mountainview, CA, U.S.A.

Appendix A

All Constraints of The Flow Based Formulation

Flow constraints for the different types of flow:

- In-segment-combination flow

Supply	$\sum_{i,j,k,l} x_{S,t,t^*,\downarrow}^{i,j,k,l,C=1,t,t^*}$	$=$	1	$\forall\ t, t^*, t \leq t^*$
FCC	$x_{S,t,t^*,\downarrow}^{i,j,k,l,C=1,t,t^*}$	$=$	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x_{i,j,k,l,C=1,t,t^*}^{\hat{i},\hat{j},\hat{k},\hat{l},C=2,t,t^*}$	$\forall\ i, j, k, l$
				$\forall\ t, t^* : t \leq t^*$
	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t,t^*}^{i,j,k,l,C,t,t^*}$	$=$	$\sum_{\hat{\hat{i}},\hat{\hat{j}},\hat{\hat{k}},\hat{\hat{l}}} x_{i,j,k,l,C,t,t^*}^{\hat{\hat{i}},\hat{\hat{j}},\hat{\hat{k}},\hat{\hat{l}},C+1,t,t^*}$	$\forall\ i, j, k, l, 1 < C < m$
				$\forall\ t, t^* : t \leq t^*$
	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x_{\hat{i},\hat{j},\hat{k},\hat{l},C=m-1,t,t^*}^{i,j,k,l,C=m,t,t^*}$	$=$	$x_{i,j,k,l,C=m,t,t^*}^{T,t,t^*,\downarrow}$	$\forall\ i, j, k, l$
				$\forall\ t, t^* : t \leq t^*$
Demand	$\sum_{i,j,k,l} x_{i,j,k,l,C=m,t,t^*}^{T,t,t^*,\downarrow}$	$=$	1	$\forall\ t, t^* : t \leq t^*$

- Horizontal flow between segments

$$
\begin{array}{lllll}
\text{Supply} & \sum_{i,j,k,l} x_{S,t^*,\rightarrow}^{i,j,k,l,C,t_1,t^*} & = & 1 & \forall\, C, t^* : t_1 < t^* - 1 \\
\text{FCC} & x_{S,t^*,\rightarrow}^{i,j,k,l,C,t_1,t^*} & = & \sum_{\hat{i},\hat{j}} x_{i,j,k,l,C,t_1,t^*}^{\hat{i},\hat{j},k,l,C,t_2,t^*} & \forall\, i,j,k,l,C \\
 & & & & \forall\, t^* : t_1 < t^* - 1 \\
 & \sum_{\hat{i},\hat{j}} x_{\hat{i},\hat{j},k,l,C,t-1,t^*}^{i,j,k,l,C,t,t^*} & = & \sum_{\hat{\hat{i}},\hat{\hat{j}}} x_{i,j,k,l,C,t,t^*}^{\hat{\hat{i}},\hat{\hat{j}},k,l,C,t+1,t^*} & \forall\, C,i,j,k,l \\
 & & & & \forall\, t,t^* : t-1 \geq t_1, t+1 < t^* \\
 & \sum_{\hat{i},\hat{j}} x_{\hat{i},\hat{j},k,l,C,t^*-2,t^*}^{i,j,k,l,C,t^*-1,t^*} & = & x_{i,j,k,l,C,t^*-1,t^*}^{T,t^*,\rightarrow} & \forall\, C,i,j,k,l \\
 & & & & \forall\, t^* : t^* \geq 3 \\
\text{Demand} & \sum_{i,j,k,l} x_{i,j,k,l,C,t^*-1,t^*}^{T,t^*,\rightarrow} & = & 1 & \forall\, C, t^* : t^* \geq 3
\end{array}
$$

- Vertical flow between segments

$$
\begin{array}{lllll}
\text{Supply} & \sum_{i,j} x_{S,t_p,C,\downarrow}^{i,j,i,j,C,t_p,t_p} & = & 1 & \forall\, C, t_p : t_p < t_H \\
\text{FCC} & x_{S,t_p,C,\downarrow}^{i,j,i,j,C,t_p,t_p} & = & \sum_{k,l} x_{i,j,i,j,C,t_p,t_p}^{i,j,k,l,C,t_p,t_p+1} & \forall\, i,j,C \\
 & & & & \forall\, t_p : t_p < t_H \\
 & \sum_{\hat{k},\hat{l}} x_{i,j,\hat{k},\hat{l},C,t_p,t^*}^{i,j,k,l,C,t_p,t^*+1} & = & \sum_{\hat{\hat{k}},\hat{\hat{l}}} x_{i,j,k,l,C,t_p,t^*+1}^{i,j,\hat{\hat{k}},\hat{\hat{l}},C,t_p,t^*+2} & \forall\, C,i,j,k,l \\
 & & & & \forall\, t_p,t^* : t_p < t_H, t^*+2 \leq t_H, t^* \geq t_p+1 \\
 & \sum_{\hat{k},\hat{l}} x_{i,j,\hat{k},\hat{l},C,t_p,t_{H-1}}^{i,j,k,l,C,t_p,t_H} & = & x_{i,j,k,l,C,t_p,t_H}^{T,t_p,C,\downarrow} & \forall\, C,i,j,k,l \\
 & & & & \forall\, t_p : t_p < t_H \\
\text{Demand} & \sum_{i,j,k,l} x_{i,j,k,l,C,t_p,t_H}^{T,t_p,C,\downarrow} & = & 1 & \forall\, C, t_p : t_p < t_H
\end{array}
$$

- Diagonal flow between segments

$$
\begin{array}{lllll}
\text{Supply} & \sum_{i,j,k,l} x_{S,\searrow}^{i,j,k,l,C,t_1,t_2} & = & 1 & \forall\, C \\
\text{FCC} & x_{S,\searrow}^{i,j,k,l,C,t_1,t_2} & = & \sum_{v,w} x_{i,j,k,l,C,t_1,t_2}^{k,l,v,w,C,t_2,t_3} & \forall\, i,j,k,l,C \\
 & \sum_{i,j} x_{i,j,k,l,C,t_p-1,t_p}^{k,l,v,w,C,t_p,t_p+1} & = & \sum_{o,q} x_{k,l,v,w,C,t_p,t_p+1}^{v,w,o,q,C,t_p+1,t_p+2} & \forall\, k,l,v,w,C \\
 & & & & \forall\, t_p : t_p - 1 \geq t_1, t_p + 2 \leq t_H \\
 & \sum_{i,j} x_{i,j,k,l,C,t_{H-2},t_{H-1}}^{k,l,v,w,C,t_{H-1},t_H} & = & x_{k,l,v,w,C,t_{H-1},t_H}^{v,w,v,w,C,t_H,t_H} & \forall\, C,k,l,v,w \\
 & \sum_{i,j} x_{i,j,k,l,C,t_{H-1},t_H}^{k,l,k,l,C,t_H,t_H} & = & x_{k,l,k,l,C,t_H,t_H}^{T,\searrow} & \forall\, C,k,l \\
\text{Demand} & \sum_{k,l} x_{k,l,k,l,C,t_H,t_H}^{T,\searrow} & = & 1 & \forall\, C
\end{array}
$$

Connection between these flows:

- in t_p, t_p segment combinations:

$$
\begin{aligned}
C = 1: \quad & x^{i,j,k,l,C=1,t_p,t_p}_{S,t_p,t_p,\downarrow} && = && x^{i,j,k,l,C=1,t_p,t_p}_{S,t_p,C,\downarrow} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_p < t_H \\
C > 1: \quad & \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_p,t_p}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_p} && = && x^{i,j,k,l,C,t_p,t_p}_{S,t_p,C,\downarrow} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_p < t_H
\end{aligned}
$$

- in the segment combination t_1, t_2:

$$
\begin{aligned}
C = 1: \quad & x^{i,j,k,l,C=1,t_1,t_2}_{S,t_1,t_2,\downarrow} && = && x^{i,j,k,l,C=1,t_1,t_2}_{S,\searrow} && \forall\ i,j,k,l \\
& x^{i,j,k,l,C=1,t_1,t_2}_{S,t_1,t_2,\downarrow} && = && x^{i,j,k,l,C=1,t_1,t_2}_{i,j,i,j,C=1,t_1,t_1} && \forall\ i,j,k,l \\
C > 1: \quad & \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_1,t_2}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_1,t_2} && = && x^{i,j,k,l,C,t_1,t_2}_{S,\searrow} && \forall\ i,j,k,l \\
& \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_1,t_2}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_1,t_2} && = && x^{i,j,k,l,C,t_1,t_2}_{i,j,i,j,C,t_1,t_1} && \forall\ i,j,k,l
\end{aligned}
$$

- in segment combinations $t_1, t_p : t_p > t_2$:

$$
\begin{aligned}
C = 1: \quad & x^{i,j,k,l,C=1,t_1,t_p}_{S,t_1,t_p,\downarrow} && = && x^{i,j,k,l,C=1,t_1,t_p}_{S,t_p,\rightarrow} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_2 < t_p \leq t_H \\
& x^{i,j,k,l,C=1,t_1,t_p}_{S,t_1,t_p,\downarrow} && = && \sum_{\hat{k},\hat{l}} x^{i,j,k,l,C=1,t_1,t_p}_{i,j,\hat{k},\hat{l},C=1,t_1,t_{p-1}} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_2 < t_p \leq t_H \\
C > 1: \quad & \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_1,t_p}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_1,t_p} && = && x^{i,j,k,l,C,t_1,t_p}_{S,t_p,\rightarrow} && \forall\ i,j,k,l \\
& && && && \forall\ t_2 < t_p \leq t_H \\
& \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_1,t_p}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_1,t_p} && = && \sum_{\hat{k},\hat{l}} x^{i,j,k,l,C,t_1,t_p}_{i,j,\hat{k},\hat{l},C,t_1,t_{p-1}} && \forall\ i,j,k,l \\
& && && && \forall\ t_2 < t_p \leq t_H
\end{aligned}
$$

- in segment combinations $t_p, t_{p+1} : t_p \geq t_2$:

$$
\begin{aligned}
C=1: \quad & x^{i,j,k,l,C=1,t_p,t_{p+1}}_{S,t_p,t_{p+1},\downarrow} && = && x^{i,j,k,l,C=1,t_p,t_{p+1}}_{i,j,i,j,C=1,t_p,t_p} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_1 < t_p < t_H \\
& x^{i,j,k,l,C=1,t_p,t_{p+1}}_{S,t_p,t_{p+1},\downarrow} && = && \sum_{\hat{i},\hat{j}} x^{i,j,k,l,C=1,t_p,t_{p+1}}_{\hat{i},\hat{j},k,l,C=1,t_{p-1},t_{p+1}} && \forall\ i,j,k,l,t_p \\
& && && && \forall\ t_1 < t_p < t_H \\
& x^{i,j,k,l,C=1,t_p,t_{p+1}}_{S,t_p,t_{p+1},\downarrow} && = && \sum_{\hat{i},\hat{j}} x^{i,j,k,l,C=1,t_p,t_{p+1}}_{\hat{i},\hat{j},i,j,C=1,t_{p-1},t_p} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_1 < t_p < t_H \\
C>1: \quad & \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_p,t_{p+1}}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_{p+1}} && = && x^{i,j,k,l,C,t_p,t_{p+1}}_{i,j,i,j,C,t_p,t_p} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_1 < t_p < t_H \\
& \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_p,t_{p+1}}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_{p+1}} && = && \sum_{\hat{i},\hat{j}} x^{i,j,k,l,C,t_p,t_{p+1}}_{\hat{i},\hat{j},i,j,C,t_{p-1},t_p} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_1 < t_p < t_H \\
& \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_p,t_{p+1}}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_{p+1}} && = && \sum_{\hat{i},\hat{j}} x^{i,j,k,l,C,t_p,t_{p+1}}_{\hat{i},\hat{j},k,l,C,t_{p-1},t_{p+1}} && \forall\ i,j,k,l \\
& && && && \forall\ t_p : t_1 < t_p < t_H
\end{aligned}
$$

- in the segment combination t_H, t_H:

$$
\begin{aligned}
C=1: \quad & x^{i,j,i,j,C=1,t_H,t_H}_{S,t_H,t_H,\downarrow} && = && \sum_{\hat{i},\hat{j}} x^{i,j,i,j,C=1,t_H,t_H}_{\hat{i},\hat{j},i,j,C=1,t_{H-1},t_H} && \forall\ i,j \\
C>1: \quad & \sum_{\hat{i},\hat{j}} x^{i,j,i,j,C,t_H,t_H}_{\hat{i},\hat{j},\hat{i},\hat{j},C-1,t_H,t_H} && = && \sum_{\hat{i},\hat{j}} x^{i,j,i,j,C,t_H,t_H}_{\hat{i},\hat{j},i,j,C,t_{H-1},t_H} && \forall\ i,j,k,l
\end{aligned}
$$

- in all remaining segment combinations t_p, t_q:

$$
\begin{aligned}
C=1: \quad & x^{i,j,k,l,C=1,t_p,t_q}_{S,t_p,t_q,\downarrow} && = && \sum_{\hat{k},\hat{l}} x^{i,j,k,l,C=1,t_p,t_q}_{i,j,\hat{k},\hat{l},C=1,t_p,t_{q-1}} && \forall\ i,j,k,l \\
& && && && \forall \text{ remaining } t_p, t_q \\
& x^{i,j,k,l,C=1,t_p,t_q}_{S,t_p,t_q,\downarrow} && = && \sum_{\hat{i},\hat{j}} x^{i,j,k,l,C=1,t_p,t_q}_{\hat{i},\hat{j},k,l,C=1,t_{p-1},t_q} && \forall\ i,j,k,l \\
& && && && \forall \text{ remaining } t_p, t_q \\
C>1: \quad & \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_p,t_q}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_q} && = && \sum_{\hat{k},\hat{l}} x^{i,j,k,l,C,t_p,t_q}_{i,j,\hat{k},\hat{l},C,t_p,t_{q-1}} && \forall\ i,j,k,l \\
& && && && \forall \text{ remaining } t_p, t_q \\
& \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x^{i,j,k,l,C,t_p,t_q}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_q} && = && \sum_{\hat{i},\hat{j}} x^{i,j,k,l,C,t_p,t_q}_{\hat{i},\hat{j},k,l,C,t_{p-1},t_q} && \forall\ i,j,k,l \\
& && && && \forall \text{ remaining } t_p, t_q
\end{aligned}
$$

The beam-on time flow is modelled by changing the in-segment-combination flow in the t_p, t_p segment combinations as follows:

$$
\begin{array}{lllll}
\text{Supply} & \sum_{i,j} x_{S,t_p,t_p,\downarrow}^{i,j,i,j,C=1,t_p,t_p} & = & 1 & \forall\ t_p \\
\text{FCC} & x_{S,t_p,t_p,\downarrow}^{i,j,i,j,C=1,t_p,t_p} & = & \sum_{\hat{i},\hat{j}} x_{i,j,i,j,C=1,t_p,t_p}^{\hat{i},\hat{j},\hat{i},\hat{j},C=2,t_p,t_p} & \forall\ i,j,\ \text{if } i=j \\
 & & & & \forall\ t_p \\
 & x_{S,t_p,t_p,\downarrow}^{i,j,i,j,C=1,t_p,t_p} & = & x_{i,j,i,j,C=1,t_p,t_p}^{T,t_p,t_p,\downarrow} & \forall\ i,j,\ \text{if } i\neq j \\
 & & & & \forall\ t_p \\
 & \sum_{\hat{i},\hat{j}} x_{\hat{i},\hat{j},\hat{i},\hat{j},C-1,t_p,t_p}^{i,j,i,j,C,t_p,t_p} & = & \sum_{\hat{\hat{i}},\hat{\hat{j}}} x_{i,j,i,j,C,t_p,t_p}^{\hat{\hat{i}},\hat{\hat{j}},\hat{\hat{i}},\hat{\hat{j}},C+1,t_p,t_p} & \forall\ i,j,C>1,\ \text{if } i=j \\
 & & & & \forall\ t_p \\
 & \sum_{\hat{i},\hat{j}} x_{\hat{i},\hat{j},\hat{i},\hat{j},C-1,t_p,t_p}^{i,j,i,j,C,t_p,t_p} & = & x_{i,j,i,j,C,t_p,t}^{T,t_p,t_p,\downarrow} & \forall\ i,j,C>1 \text{ if } i\neq j \\
 & & & & \forall\ t_p \\
 & \sum_{\hat{i},\hat{j}} x_{\hat{i},\hat{j},\hat{i},\hat{j},C=m-1,t_p,t_p}^{i,j,i,j,C=m,t_p,t_p} & = & x_{i,j,i,j,C=m,t_p,t_p}^{T,t_p,t_p,\downarrow} & \forall\ i,j,t_p \\
\text{Demand} & \sum_{C,i,j,k,l} x_{i,j,i,j,C,t_p,t_p}^{T,t_p,t_p,\downarrow} & = & 1 & \forall\ t_p
\end{array}
$$

Additionally we have to change the matching of flows in the t_p, t_p ($t_p < t_H$) segment combinations

$$
\begin{array}{lllll}
C=1: & x_{S,t_p,t_p,\downarrow}^{i,j,k,l,C=1,t_p,t_p} & = & x_{S,t_p,C,\downarrow}^{i,j,k,l,C=1,t_p,t_p} & \forall\ i,j,k,l,t_p : t_p < t_H \\
C>1: & \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_p}^{i,j,k,l,C,t_p,t_p} & \leq & x_{S,t_p,C,\downarrow}^{i,j,k,l,C,t_p,t_p} & \forall\ i,j,k,l \\
 & & & & \forall\ t_p : t_p < t_H
\end{array}
$$

and in the segment combination t_H, t_H:

$$
\begin{array}{lllll}
C=1: & x_{S,t_H,t_H,\downarrow}^{i,j,i,j,C=1,t_H,t_H} & = & \sum_{\hat{i},\hat{j}} x_{\hat{i},\hat{j},i,j,C=1,t_{H-1},t_H}^{i,j,i,j,C=1,t_H,t_H} & \forall\ i,j \\
C>1: & \sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x_{\hat{i},\hat{j},\hat{i},\hat{j},C-1,t_H,t_H}^{i,j,i,j,C,t_H,t_H} & \leq & \sum_{\hat{i},\hat{j}} x_{\hat{i},\hat{j},i,j,C,t_{H-1},t_H}^{i,j,i,j,C,t_H,t_H} & \forall\ i,j
\end{array}
$$

The flow, which models the setup time, is realized by increasing the supply and the demand in the internal sinks and sources, respectively, of the segment combinations t_p, t_{p+1}. Additionally, we double each arc in this segment, which is connecting two internal vertices. Hence, we get the following additional set of constraints:

Supply	$\sum_{i,j,k,l} \hat{x}_{S,t_p,t_{p+1},\downarrow}^{i,j,k,l,C=1,t_p,t_{p+1}}$	$=$	1	$\forall\ t_p : t_p < t_H$
Constr	$\hat{x}_{S,t_p,t_{p+1},\downarrow}^{i,j,k,l,C=1,t_p,t_{p+1}}$	$=$	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} \hat{x}_{i,j,k,l,C=1,t_p,t_{p+1}}^{\hat{i},\hat{j},\hat{k},\hat{l},C=2,t_p,t_{p+1}}$	$\forall\ i,j,k,l : i = k, j = l$
	$\hat{x}_{S,t_p,t_{p+1},\downarrow}^{i,j,k,l,C=1,t_p,t_{p+1}}$	$=$	$\hat{x}_{i,j,k,l,C=1,t_p,t_{p+1}}^{T,t_p,t_{p+1},\downarrow}$	$\forall\ t_p : t_p < t_H$ $\forall\ i,j,k,l : i \neq k$ or $j \neq l$
	$\hat{x}_{S,t_p,t_{p+1},\downarrow}^{i,j,k,l,C=1,t_p,t_{p+1}}$	$=$	$x_{S,t_p,t_{p+1},\downarrow}^{i,j,k,l,C=1,t_p,t_{p+1}}$	$\forall\ t_p : t_p < t_H$ $\forall\ i,j,k,l$
	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} \hat{x}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_{p+1}}^{i,j,k,l,C,t_p,t_{p+1}}$	$\leq$	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} x_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_{p+1}}^{i,j,k,l,C,t_p,t_{p+1}}$	$\forall\ t_p : t_p < t_H$ $\forall\ i,j,k,l,C > 1, t_p : t_p < t_H$
	$\hat{x}_{i,j,k,l,C,t_p,t_{p+1}}^{T,t_p,t_{p+1},\downarrow}$	$=$	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} \hat{x}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_{p+1}}^{i,j,k,l,C,t_p,t_{p+1}}$	$\forall i,j,k,l,C > 1, i \neq k$ or $j \neq l$
	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} \hat{x}_{\hat{i},\hat{j},\hat{k},\hat{l},C-1,t_p,t_{p+1}}^{i,j,k,l,C,t_p,t_{p+1}}$	$=$	$\sum_{\hat{\hat{i}},\hat{\hat{j}},\hat{\hat{k}},\hat{\hat{l}}} \hat{x}_{i,j,k,l,C,t_p,t_{p+1}}^{\hat{\hat{i}},\hat{\hat{j}},\hat{\hat{k}},\hat{\hat{l}},C+1,t_p,t_{p+1}}$	$\forall\ t_p : t_p < t_H$ $\forall\ i = k, j = l, 1 < C < m$
	$\sum_{\hat{i},\hat{j},\hat{k},\hat{l}} \hat{x}_{\hat{i},\hat{j},\hat{k},\hat{l},C=m-1,t_p,t_{p+1}}^{i,j,k,l,C=m,t_p,t_{p+1}}$	$=$	$\hat{x}_{i,j,k,l,C=m,t_p,t_{p+1}}^{T,t_p,t_p,\downarrow}$	$\forall\ t_p : t_p < t_H$ $\forall\ i,j,k,l$
Demand	$\sum_{C,i,j,k,l} \hat{x}_{i,j,k,l,C,t_p,t_{p+1}}^{T,t_p,t_p,\downarrow}$	$=$	1	$\forall\ t_p : t_p < t_H$ $\forall\ t_p : t_p < t_H$

Figures

Bibliography

[Anbil et al.] R. Anbil, R. Tanga, E. L. Johnson, "A global approach to crew-pairing optimizaton", *IBM Systems Journal*, Vol. **31**, p 71 - 78, (1992)

[Atamtürk et al.] A. Atamtürk, G. L. Nemhauser, M. W. P. Savelsbergh, "A combined lagrangian, linear programming and implication heuristic for large-scale set partitioning problems", *Journal of Heuristics*, Vol. **1**, p 247 - 259, (1995)

[Atlas of cancer mortality] N. Becker, J. Wahrendorf, "Atlas of cancer mortality in the Federal Republic of Germany 1981 - 1990", 3rd edition, Springer, Berlin, (1998)

[Balas & Padberg] E. Balas, M. W. Padberg, "Set partitioning: a survey", *SIAM Review*, Vol. **18**, p 710 - 760, (1976)

[Barahona& Anbil] F. Barahona, R. Anbil, "The volume algorithm: producing primal solutions with a subgradient method", *IBM Research Report*, Vol. **RC 21103**, (1998)

[Barnhart, Johnson et al.] C. Barnhart, E. J. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, P. H. Vance, "Branch-and-price: column generation for solving huge integer programs", *Operations Research*, Vol. **46**, p 316 - 329, (1998)

[Borndörfer] R. Borndörfer, "Aspects of set packing, partitioning and covering", Shaker Verlag, Aachen, (1998)

[Bortfeld et al.(1994)] Th. Bortfeld, D. L. Kahler, T. J. Waldron, A. L. Boyer, "X-ray Field Compensation with Multileaf Collimators", *International Journal of Radiation Oncology Biology Physics*, Vol. **28**, p 723 - 730, (1994)

[Bortfeld (1995)] Th. Bortfeld, "Dosiskonformation in der Tumortherapie mit externer ionisierender Strahlung: Physikalische Möglichkeiten und Grenzen", Habilitationsschrift, Deutsches Krebsforschungszentrum, Heidelberg, (1995)

[Convery & Rosenbloom] D. J. Convery, M. E. Rosenbloom, "The generation of intensity-modulated fields for conformal radiotherapy by dynamic collimation", *Physics in Medicine and Biology*, Vol. **37**, p 1359 - 1374, (1992)

[Galvin et al.(1993)] J. M. Galvin, X.-G. Chen, R. M. Smith, "Combining Multileaf Fields to modulate Fluence Distributions", *International Journal of Radiation Oncology Biology Physics*, Vol. **27**, p 697-705, (1993)

[Garey & Johnson (1979)] M. R. Garey, D. S. Johnson, "Computers and intractability : a guide to the theory of NP-Completeness", Freeman, San Francisco, (1979)

[Hamacher & Küfer (1999)] H. W. Hamacher, K.-H. Küfer, "Inverse Radiation Therapy Planning - a Multiple Objective Optimisation Approach", *Berichte des ITWM*, Vol. **12**, (1999), (to appear in *Discrete Applied Mathematics*)

[Kim & Pardalos (1999)] D. Kim, P. M. Pardalos, "A solution approach to the fixed charge network problem using a dynamic slope scaling procedure", *Operations Research Letters*, Vol. **24**, p 195-203, (1999)

[Lasdon] L. S. Lasdon, "Optimization Theory for Large Systems", MacMillan Publishing Co., Inc., New York, (1970)

[Marsten] R. Marsten, "Crew planning at Delta Airlines", *Presentation at Mathematical Programming Symposium XV*, Ann Arbor, MI., (1994)

[Mohan (1995)] R. Mohan, "Field Shaping for Three-Dimensional Conformal Radiation Therapy and Multileaf Collimators", *Seminars in Radiation Oncology*, Vol. **5**, p 86-99, (1995)

[Papatheodorou et al.] S. Papatheodorou, J. C. Rosenwald, M. E. Castellanos, S. Zefkili, L. Bonvalet, G. Gaboriaud, "Utilisation d'un collimateur multilames pour la production de faisceaux modulés en intensité", *Cancer Radiothérapie*, Vol. **2**, p 392 - 403, (1998)

[Peregrine] `http://www-phys.llnl.gov/peregrine`

[Preiser et al. (1998)] K. Preiser, Th. Bortfeld, K. Hartwig, W. Schlegel, J. Stein, "Inverse Strahlentherapieplanung für intensitätsmodulierte Photonenfelder", *Der Radiologe*, Vol. **38**, p 228-234 , (1998)

[van Santvoort & Heijmen] J. P. C. van Santvoort, B. J. M. Heijmen, "Dynamic multileaf collimation without 'tongue-and-groove' underdosage effects", *Physics in Medicine and Biology*, Vol. **41**, p 2091 - 2105, (1996)

[Siochi (1999)] R. A. C. Siochi, "Minimizing static intensity modulation delivery time using an intensity solid paradigm.", *International Journal of Radiation Oncology Biology Physics*, Vol. **43**, p 671 - 680, (1999)

[Svensson et al.] R. Svensson, P. Källman, A. Brahme, "An analytical solution for the dynamic control of multileaf collimators", *Physics in Medicine and Biology*, Vol. **39**, p 37 - 61, (1994)

[Ribeiro et al.] C. C. Ribeiro, M. Minoux, M. C. Penna, "An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment", *European Journal of Operational Research*, Vol. **41**, p 232 - 239, (1989)

[Webb (1998a)] S. Webb, "Configuration options for intensity-modulated radiation therapy using multiple static fields shaped by a multileaf collimator", *Physics in Medicine and Biology*, Vol. **43**, p 241 - 260, (1998)

[Webb (1998b)] S. Webb, "Configuration options for intensity-modulated radiation therapy using multiple static fields shaped by a multileaf collimator. II: Constraints and limitations on 2D modulation", *Physics in Medicine and Biology*, Vol. **43**, p 1481 - 1495, (1998)

[Xia & Verhey (1998)] P. Xia, L. J. Verhey, "Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments", *Medical Physics*, Vol. **25**, p 1424 - 1434, (1998)

[Yu et al.] C. X. Yu, D. Yan, M. N. Du, S. Zhou, L. J. Verhey, "Optimizaion of leaf-positions when shaping a radiation field with a multileaf collimator", *Physics in Medicine and Biology*, Vol. **40**, p 305 - 308,(1995)

Declaration

I hereby declare that I am the only author of this work and that no other sources than those listed have been used.

Kaiserslautern, Germany, June 30, 2000

Frank Lenzen