

Davaatseren Baatar

Matrix Decomposition with Time and Cardinality Objectives:

Theory, Algorithms and Application to Multileaf Collimator Sequencing

Von Fachbereich Mathematik
der Universität Kaiserslautern
genehmigte

Dissertation

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
(Doctor rerum naturalium. Dr. rer. nat.)

Gutachter: Prof. Dr. Horst W. Hamacher
Gutachter: Prof. Dr. Natasha Boland

Datum der Disputation: 25. August 2005

D 386

Acknowledgements

This thesis was made possible with help, advice, support and feedback from many people.

First and foremost, I would like to express my endless gratitude to my supervisor Prof. Horst W. Hamacher, who was encouraging, perceptive, helpful and supportive in ways too numerous to mention. Thank you very much for your patience.

My gratitude continues to Prof. Natasha Boland, University of Melbourne, for her good will and effort to read and evaluate my thesis. Sorry for weekends spent reading my work.

I am also grateful to Dr. Matthias Ergott for the warm cooperation and useful discussions which helped me enormously .

Many thanks for all my colleagues at AG Hamacher for their friendly support for all seasons.

The financial support of the Graduiertenkolleg "Mathematik und Praxis" gratefully acknowledged for financing my stay during the research.

Last but not least loving and special thanks to my wonderful family, my wife Bulganchimeg, daughters Indra and Amina. Without your love, patience, support I am lost.

Contents

1	Introduction	6
1.1	Intensity Modulated Radiation Therapy	6
1.2	Multileaf collimator and related Optimization Problems	10
2	Decomposability	15
2.1	Necessary and Sufficient Condition	15
2.2	Invariant Property	19
2.3	Decomposability on a Given Set	22
3	Decomposition Time Problem	25
3.1	Compact Model For Decomposition Time Problem	25
3.2	Totally unimodularity	26
3.3	Algorithm and Numerical results	30
3.4	Nested sequence of Multicriteria Optimization problems	31
3.5	Upper Bound on End Positions	38
3.6	Unconstrained Decomposition Time Problem	45
4	Decomposition Cardinality Problem	47
4.1	NP-hardness	47
4.2	Polynomially solvable Cases	51
4.3	IP model	52
4.4	Ordering, Fixing and Line search scheme	56
4.5	Numerical results	57
4.6	Bender's Combinatorial Cut	58
4.7	Decomposition	59
4.8	Strengthen the Master Problem	62
4.9	Irreducible Inconsistent Subset	63
4.10	Numerical result	64

5	Heuristics	66
5.1	DT Solution Based Greedy Heuristic	66
5.1.1	Constrained DC	66
5.1.2	Unconstrained case	71
5.2	Numerical results	74
5.3	Flexible Minimal DT Based Heuristic	75
5.4	Numerical Result	79
6	Conclusion	82
	Bibliography	83

Chapter 1

Introduction

1.1 Intensity Modulated Radiation Therapy

Every year more than 340000 people in Germany are diagnosed with cancer and more than 210000 of them die [6]. At the time of being diagnosed about 60% of all cancer patients are suffering from malignant localized tumor which has not yet disseminated, i.e., no metastatic disease has yet occurred. Thus these patients can be considered to be potentially curable. Nevertheless, about one third of these patients (18% of all cancer patients) cannot be cured, because therapy fails to stop tumor growth. This fact shows that the success in cancer treatment is far from being satisfactory at the moment.

Radiotherapy is, after surgery, the most successfully and most frequently used treatment modality for cancer. Radiotherapy aims to deliver a radiation dose to the tumor, so that the cells are destroyed or at least be controlled in its growth. The fact that tumor surrounding organs- the organs at risk- are in general very sensitive to radiation is a pressing problem. The situation is even more complicated when the tumor itself is rather radio resistant and very high doses are needed to achieve a therapeutic effect.

In recent years, advances in radiation therapy have led to the development of delivery techniques with a high degree of computer control. These techniques offer many new opportunities in the delivery of radiation therapy. Due to the complexity of the treatments, however, an optimization approach is needed in order to develop the best plan for treating each patients.

Medical Linear Accelerators (commonly called linacs) are today widely used and are the most common units for treating deep seated tumors by external irradiation. The basic idea of an electronlinac is to accelerate electrons in the field of an electromagnetic wave traveling in a waveguide. They are put in a

gantry, which can be rotated around the patient, who is fixed on a couch, and deliver radiation from different locations (See Figure 1.1).



Figure 1.1: A medical linear accelerator with beam head and a couch. A patient is being treated.

The cut through the beam's head is shown in Figure 1.2, which shows how radiation is focused, filtered and a cone shaped field is created toward the target volume. Typically there are 36 positions, equally distributed on the gantry's moving circle around the patient. There is no restriction on the number of positions to be used. But in clinical practice a number between 3-7 is most common.

Figure 1.3 represents a 2D slice of the human body and the cone shaped radiation delivered from different positions toward the target volume. The view from the radiation source is called beam's eye view.

The objectives of treatment planning, high level dose in the tumor and low radiation in the organs at risk, have a basically contradictory nature. In most hospitals, radiation treatment planning is performed with a *design-reuse* or *forward strategy*, i.e., an old treatment set-up applied in a clinical related case is reused for the incoming patient. The experienced planner will modify the old set-up parameters by trial and error until a *suitable* therapy for the current patient is achieved. It is no surprise that for complicated anatomical situations, in particular in case of none convex shapes of the target volume, the forward strategy is time consuming and produces unsatisfactory results. A "*suitable*" therapy plan is all too often far from being the best possible [25].

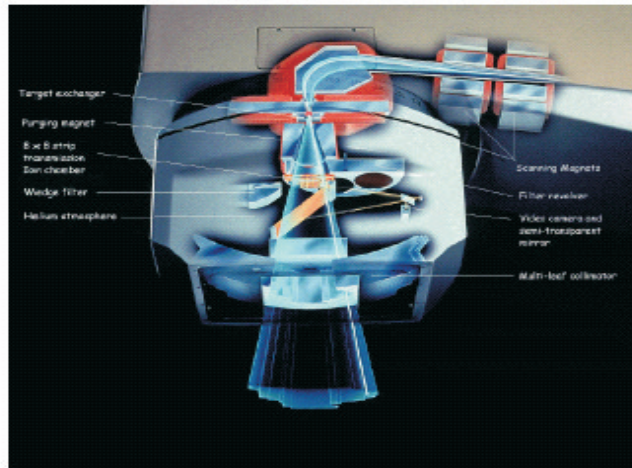


Figure 1.2: A profile of the beam's head.

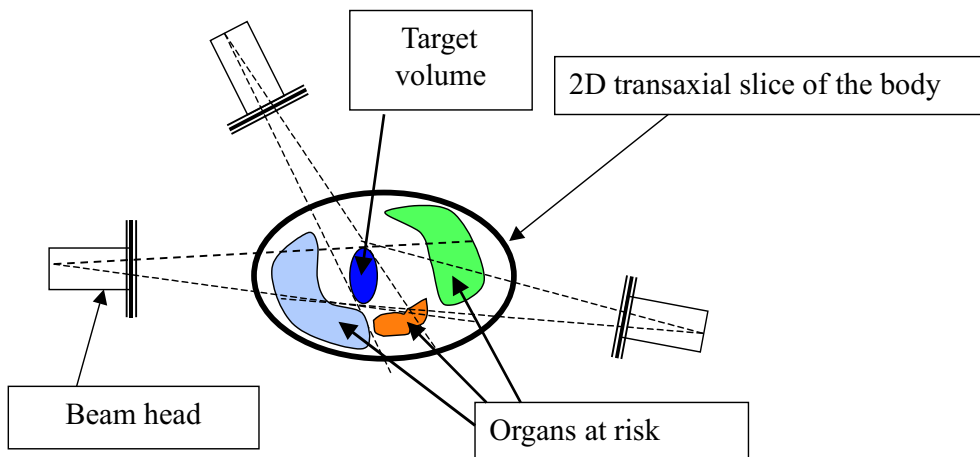


Figure 1.3: Radiation treatment from 3 locations.

For this reason, treatment planning problems in recent years have been modelled using an *inverse* or *backward strategy*. Let the desired dose bounds be given: high dose level on target volume and avoiding overdose in relevant healthy tissue nearby. The problem is to determine the intensity distribution in the treatment space in such a way that the prescribed dose distribution in the patient space is obtained. Inverse treatment planning traditionally faces 2 inverse problems. The problem just mentioned above is the first inverse problem. For this problem many inverse treatment algorithms have been developed [25, 59].

5	2	5	0
5	10	2	0
2	0	8	0
0	3	3	8

Figure 1.4: Intensity map(matrix) with dimension 4x4.

Intensity modulation is a powerful tool for creating dose distributions that spare critical organs and that have high dose regions conforming to the tumor shape. This technique improves dose conformity by allowing the intensity of the beam to vary across the patient surface. This variation can be visualized as an intensity map, Figure 1.4, a 2D matrix whose cells have a one-to-one correspondence with the bixels Figure 1.5 and with the patient coordinates in

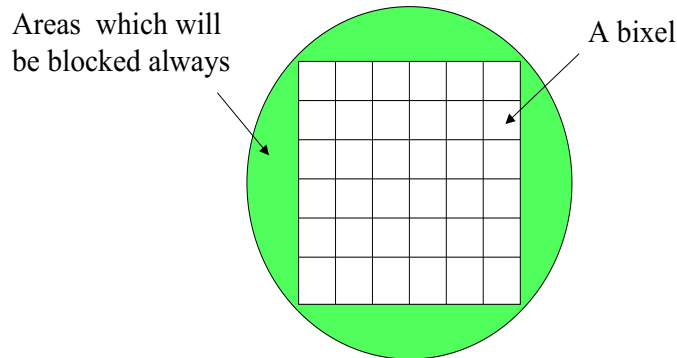


Figure 1.5: Discretized area seen from the beam's eye. A cell of this area is called bixel.

the iso-centric plane and whose entries are directly proportional to the amount of time that the corresponding cell of the matrix is exposed to radiation. The entries in the map are all integers, and a scaling factor multiplied to the map

is used to determine the number of monitor units each cell receives. Inverse treatment planning algorithms generate the intensity maps, and these maps could be delivered with techniques which involves *dynamic* Multileaf Collimator (MLC), or *static* MLC.

1.2 Multileaf collimator and related Optimization Problems

In order to deliver an intensity map, the uniform radiation leaving the linear accelerator has to be modulated by inserting filters on its way to the patient. A more advanced way of modulation is achieved by using a MLC (Figure 1.6).



Figure 1.6: The adjacent leaf pairs in MLC.

MLC consists of high metal pieces so called leaves, usually 5–7 cm high and 5–10 mm thick, which can block the radiation. Pairs of leaves are placed adjacent to each other. Each leaf in a pair can move in the direction toward the other leaf or away from it. Each row of intensity map, often referred to as a *channel*, has an associated pair of leaves – a right leaf and a left leaf. Different shapes can be given to the radiation by changing the positions of the leaves in the channels. We can represent any leaf configurations by using binary matrices where 1's present the bixels exposed to radiation and 0's present the bixels blocked by leaves.

Definition 1 A binary matrix is called a **shape matrix** if it represents leaf

configurations.

These binary matrices have so called consecutive one's property:

Definition 2 A binary matrix is a (strict) **consecutive ones matrix**, or **C1 matrix** for short, if the ones occur consecutively in a single block in each row.

But definition of shape matrices is dependent on different models of multileaf collimators. For some type of MLC, for example VARIAN MLC, any C1 matrix is a shape matrix. On the other hand, for some MLC a leaf motion is restricted, like in a case of Siemens MLC, where a leaf is not allowed to move further than its opposing leaf and opposing leaves in adjacent channels. These restrictions on the motion of the leaves is called *interleaf motion constraints* or *leaf collision constraints*. In order to represent leaf configurations of such MLC we need a certain type of binary matrices with respect to the leaf motion constraints i.e. a C1 matrix $Y = (y_{mn})_{M \times N}$ is a shape matrix if there exist ℓ_m and r_m , for all rows $m = 1, \dots, M$, such that

$$\begin{aligned} y_{mn}^k = 1 &\iff \ell_m^k \leq n < r_m^k \quad \forall m = 1, \dots, M \\ \ell_{m-1} &\leq r_m \quad \forall m = 2, \dots, M \\ r_{m-1} &\geq \ell_m \quad \forall m = 2, \dots, M \end{aligned}$$

In this thesis we just use the term shape matrix to present leaf configurations and from the context it will be clear which kind of C1 matrices we are considering.

The methods used to control the collimator leaves can be divided into 2 categories:

- In *Dynamic* MLC techniques the leaves can be moved during the delivery of the radiation and so the dose is delivered continuously corresponding to the intensity map
- In *Static* MLC techniques the positions of the leaves are discretely changed only when the irradiation is interrupted (i.e. beam is off during the motion of the leaves)

The second inverse problem of radiation treatment planning utilizing the MLC is to determine the sequencing of the leaves to produce the required intensity map. This thesis will focus on the second inverse problem. We will consider the problem only for static MLC.

Mathematically, we are looking for a "suitable" presentation of the intensity map A as a positive linear combination of shape matrices. Let Ω be an index set of all $M \times N$ shape matrices. We consider the following problem. Given an $M \times N$ matrix $A = (a_{m,n})$ with non-negative integer entries, find a "good" **C1**

decomposition, i.e. non-negative integers $\alpha_k, k \in \Omega$ and $M \times N$ C1 matrices $Y^k, k \in \Omega$ such that

$$A = \sum_{k \in \Omega} \alpha_k Y^k. \quad (1.1)$$

In the following, we often use $\mathcal{M} := \{1, \dots, M\}$, $\mathcal{N} = \{1, \dots, N + 1\}$.

For each of the C1 matrices Y^k there exist $\ell_m^k \in \mathcal{N}, r_m^k \in \mathcal{N}$ such that $Y^k = (y_{mn}^k)$ is given by

$$y_{mn}^k = 1 \iff \ell_m^k \leq n < r_m^k \quad \forall m \in \mathcal{M}. \quad (1.2)$$

Using $[p, q) := \{i \in \mathcal{N} : p \leq i < q\}$ C1 matrices Y^k can be written as

$$Y^k = Y([\ell_m^k, r_m^k))_{m \in \mathcal{M}}.$$

Example 1.2.1 For $A = \begin{pmatrix} 2 & 5 & 3 \\ 3 & 5 & 2 \end{pmatrix}$,

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

is a possible decomposition defined by

$$\begin{aligned} \ell^1 &= \begin{pmatrix} 1 \\ 1 \end{pmatrix}, & \ell^2 &= \begin{pmatrix} 2 \\ 1 \end{pmatrix}, & \ell^3 &= \begin{pmatrix} 2 \\ 2 \end{pmatrix}; \\ r^1 &= \begin{pmatrix} 4 \\ 4 \end{pmatrix}, & r^2 &= \begin{pmatrix} 4 \\ 3 \end{pmatrix}, & r^3 &= \begin{pmatrix} 3 \\ 3 \end{pmatrix}; \\ \alpha_1 &= 2 & \alpha_2 &= 1 & \alpha_3 &= 2. \end{aligned}$$

The representation of Y^2 in terms of intervals is $Y^2 = Y \begin{pmatrix} [2, 4) \\ [1, 3) \end{pmatrix}$.

It should be noted that the definition of C1 matrices is usually more general than ours: Any 0-1 matrix which can be transformed by column permutations into a matrix where all ones occur consecutively in the rows (see, e.g., [9]). For this reason our definition contains the word strict which we will, however, delete subsequently.

The word "suitable" can present different goals as:

- **Minimal beam-on time(Decomposition time problem)**

$$\begin{aligned}
DT(\alpha) = \min & \sum_{t \in \Omega} \alpha_t \\
\text{s.t.} & \sum_{t \in \Omega} \alpha_t * S_t = A \\
& \alpha_t \geq 0 \quad \forall t \in \Omega
\end{aligned}$$

- **Minimal number of shape matrices(Decomposition Cardinality Problem)**

$$\begin{aligned}
DC(\alpha) = \min & |\mathcal{T}| \\
\text{s.t.} & \sum_{t \in \mathcal{T}} \alpha_t * S_t = A \\
& \alpha_t \geq 0 \quad \forall t \in \mathcal{T} \\
& \mathcal{T} \subseteq \Omega
\end{aligned}$$

where Ω is the index set of all possible shape matrices.

From now on, we refer to the constraints $\sum_{t \in \mathcal{T}} \alpha_t * S_t = I$ as an "intensity cover" constraint.

If shape matrix presents a C1 matrix then we call the related problem as an unconstrained problem. If the shape matrix is defined with respect to leaf collision constraints then we call the related problem as a "constrained problem".

In radiation treatment planning it is desirable to have a solution which minimizes both DT and DC. Such a solution exists for a single row intensity matrix. However, to find this solution is strongly NP-hard problem (4.1). For 2D case, in general, there is no such a solution exists (see Example 4.1.1).

A number of authors have considered these problems and introduced many different models and heuristics based on different approaches. Some algorithms aim at solving DT problems optimally, however in such cases DC tends to increase. On the other hand some algorithms aim at decreasing DC, though DT is considerably high. There is also another approach which tries to balance those two contradictory tendencies.

In 1994 Bortfeld [10] introduced an algorithm for unconstrained DT problem. The algorithm forced leaves to move in one direction from left to right and a leaf stops at a certain position if and only the intensity of the bixel, which is next to that position, is not yet satisfied. This algorithm solves the unconstrained DT problem to optimality. However, he did not provide any proof of optimality of his algorithm. Later on, Baatar et al [4], Hamacher and Ahuja [1] and Kamath [33] in their works showed the optimality of the sweep technique. In this case the decomposition cardinality is considerably high.

In 1998 Xia and Verhey [65] introduced heuristic algorithm for DC problem. The algorithm decomposes the given intensity matrix into a set of integer multiple

of binary matrices using power of 2. Then for each binary matrix they used a heuristic algorithm to decompose it into sum of shape matrices. Though this approach yields small decomposition cardinality, it leads to long decomposition time. In 1999 Siochi [53] introduced heuristic algorithm which produces results that compromise the two objectives.

Several other authors considered these problems separately. In 2003 Boland, Hamacher, Lenzen [8] considered constrained DT problem as a network flow problem with side constraints. They showed that the constrained DT problem can be solved using column generation rule and the subproblem can be solved as a shortest path problem on a network. Moreover they proved that DT problem can be solved in polynomial time. In the paper of Langer et al [35] new MIP formulation of the problems was introduced. However, this approach was computationally not efficient. In 2003 Baatar and Hamacher [3], Kalinowsky [32] introduced new polynomial time algorithms for constrained DT problems and heuristics for DC problem.

For unconstrained DT problems Ahuja and Hamacher [1] and Kamath [33] introduced exact algorithms. They have mathematically proved that their algorithms yields exact optimal solution in polynomial time.

A part of this thesis is published in Proceedings of the Operations Research Peripatetic Postgraduate Programme Conference ORP³, Lambrecht, Germany [3], and accepted for publication in Discrete Applied Mathematics [4]. In Chapter 2 we will explore properties of the integer decompositions and based on the theoretical results new formulation is developed in Chapter 3 and Chapter 4 for DT and DC problems respectively. Then in Chapter 5 we will introduce heuristic algorithms which yields the best solutions among other existing algorithms and that is computationally efficient.

Chapter 2

Decomposability

In this chapter we explore necessary and sufficient conditions to be satisfied such that given matrix can be decomposed with certain decomposition time or on a set of predefined shape matrices. Those conditions are the primary in terms of developing models for our problems in the next chapters. We focus on the constrained case where the leaf motion is restricted, and theoretical results obtained can be adopted to the unconstrained problem just by deleting the constraints associated with interleaf motion restrictions.

2.1 Necessary and Sufficient Condition

Recall that, for a constrained case C1 matrix $Y = Y([\ell_m, r_m])_{m \in \mathcal{M}}$ is called a **shape matrix** if

$$\ell_{m-1} \leq r_m \quad \text{and} \quad r_{m-1} \geq \ell_m$$

holds for all $m = 2, \dots, M$. For a given matrix $A = (a_{mn})_{\substack{m=1, \dots, M \\ n=1, \dots, N}}$, we define the $M \times (N + 1)$ **difference matrix** $\tilde{A} = (\tilde{a}_{mn})_{\substack{m \in \mathcal{M} \\ n \in \mathcal{N}}}$ by

$$\tilde{a}_{mn} := a_{mn} - a_{m, n-1} \quad \text{for all } m \in \mathcal{M}, n \in \mathcal{N}. \quad (2.1)$$

Here $a_{m0} = a_{m, n+1} := 0$ for all $m \in \mathcal{M}$. Assume that we are looking for a decomposition with predefined decomposition time DT. The next theorem gives the answer whether it is possible to find such a decomposition or not.

Theorem 1 *A has a decomposition with decomposition time $DT(\alpha)$ if and only if there exist $M \times (N + 1)$ matrices $L = (\alpha_{mn}^\ell)$ and $R = (\alpha_{mn}^r)$ with non-negative*

elements such that

$$L - R = \tilde{A} \quad (2.2)$$

$$\sum_{k=1}^n \alpha_{m-1,k}^\ell \geq \sum_{k=1}^n \alpha_{mk}^r \quad \forall m \in \mathcal{M} \setminus \{1\}, \forall n \in \mathcal{N} \quad (2.3)$$

$$\sum_{k=1}^n \alpha_{mk}^\ell \geq \sum_{k=1}^n \alpha_{m-1,k}^r \quad \forall m \in \mathcal{M} \setminus \{1\}, \forall n \in \mathcal{N} \quad (2.4)$$

$$DT(\alpha) = \sum_{k \in \mathcal{K}'} \alpha_k = \sum_{n \in \mathcal{N}} \alpha_{pn}^\ell = \sum_{n \in \mathcal{N}} \alpha_{mn}^r \quad \forall p, m \in \mathcal{M} \quad (2.5)$$

Proof:

“ \Rightarrow ” Let a decomposition

$$A = \sum_{k \in \mathcal{K}'} \alpha_k Y^k$$

be given. We consider the matrices $L = (\alpha_{mn}^\ell)$ and $R = (\alpha_{mn}^r)$ obtained by

$$\alpha_{mn}^\ell = \sum_{k: \ell_m^k = n} \alpha_k,$$

$$\alpha_{mn}^r = \sum_{k: r_m^k = n} \alpha_k.$$

It is clear that (2.5) holds. From the element wise presentation of the decomposition

$$a_{mn} = \sum_{k: \ell_m^k < n < r_m^k} \alpha_k + \sum_{k: \substack{\ell_m^k = n \\ r_m^k > n}} \alpha_k,$$

$$a_{m,n-1} = \sum_{k: \ell_m^k < n < r_m^k} \alpha_k + \sum_{k: \substack{\ell_m^k < n \\ r_m^k = n}} \alpha_k$$

we get

$$\begin{aligned} a_{mn} - a_{m,n-1} &= \sum_{k: \substack{\ell_m^k = n \\ r_m^k > n}} \alpha_k - \sum_{k: \substack{\ell_m^k < n \\ r_m^k = n}} \alpha_k \\ &= \left(\sum_{k: \substack{\ell_m^k = n \\ r_m^k > n}} \alpha_k + \sum_{k: \substack{\ell_m^k = n \\ r_m^k = n}} \alpha_k \right) - \left(\sum_{k: \substack{\ell_m^k < n \\ r_m^k = n}} \alpha_k + \sum_{k: \substack{\ell_m^k = n \\ r_m^k = n}} \alpha_k \right) \\ &= \sum_{k: \ell_m^k = n} \alpha_k - \sum_{k: r_m^k = n} \alpha_k = \alpha_{mn}^\ell - \alpha_{mn}^r. \end{aligned}$$

Thus L and R satisfy (2.2).

For any $m \in \mathcal{M}$ and $n \in \mathcal{N}$, with $m \geq 2$, consider the set of shape matrices used in a decomposition with intervals $[\ell_m^k, r_m^k)$ where $r_m^k \leq n$. By definition of shape matrices each interval $[\ell_m^k, r_m^k)$ in a shape matrix Y^k has a corresponding $[\ell_{m-1}^k, r_{m-1}^k)$ in row $m-1$ such that

$$\ell_{m-1}^k \leq r_m^k.$$

Hence $\{Y^k : r_m^k \leq n\} \subseteq \{Y^k : \ell_{m-1}^k \leq n\}$ and we conclude

$$\sum_{k:r_m^k \leq n} \alpha_k \leq \sum_{k:\ell_{m-1}^k \leq n} \alpha_k.$$

Consequently, we get

$$\sum_{k=1}^n \alpha_{m-1,k}^\ell \geq \sum_{k=1}^n \alpha_{mk}^r,$$

i.e., conditions (2.3) holds. Using a similar observation with $\ell_m^k \leq r_{m-1}^k$ we can derive (2.4).

“ \Leftarrow ” Let L and R be matrices such that (2.2) – (2.5) hold. Let $\alpha_{m\ell_m}^\ell$ and $\alpha_{mr_m}^r$, $m \in \mathcal{M}$ be the first non-zero elements in the rows of matrices L and R , respectively, i.e.,

$$\begin{aligned} 1 \leq n < \ell_m &\Rightarrow \alpha_{mn}^\ell = 0 \quad \text{and} \quad \alpha_{m\ell_m}^\ell > 0, \\ 1 \leq n < r_m &\Rightarrow \alpha_{mn}^r = 0 \quad \text{and} \quad \alpha_{mr_m}^r > 0. \end{aligned}$$

From (2.2) we get for all $n \in \mathcal{N}$

$$\sum_{k=1}^n \alpha_{mk}^\ell - \sum_{k=1}^n \alpha_{mk}^r = a_{mn} \tag{2.6}$$

and $a_{mn} \geq 0$. Therefore,

$$\sum_{k=1}^n \alpha_{mk}^\ell \geq \sum_{k=1}^n \alpha_{mk}^r \quad \forall n \in \mathcal{N} \tag{2.7}$$

yields that

$$\ell_m \leq r_m \quad \forall m \in \mathcal{M}.$$

Moreover, from (2.3) and (2.4) it follows that

$$\begin{aligned} \ell_{m-1} &\leq r_m, \\ \ell_m &\leq r_{m-1} \end{aligned}$$

for all $m \in \mathcal{M} \setminus \{1\}$.

Therefore,

$$Y^1 = Y \begin{pmatrix} [\ell_1, r_1) \\ [\ell_2, r_2) \\ \vdots \\ [\ell_M, r_M) \end{pmatrix}$$

is a shape matrix. We choose

$$\alpha_1 = \min \{ \alpha_{1\ell_1}^\ell, \dots, \alpha_{M\ell_M}^\ell, \alpha_{1r_1}^r, \dots, \alpha_{Mr_M}^r \}.$$

Replacing $\alpha_{m\ell_m}^\ell$ and $\alpha_{mr_m}^r$, $m = 1, \dots, M$, by $\alpha_{m\ell_m}^\ell - \alpha_1$ and $\alpha_{mr_m}^r - \alpha_1$ in L and R , respectively, we get matrices L' and R' which satisfy again (2.3), (2.4) and (2.7).

Thus by repeating the above procedure until the matrices L and R simultaneously become zero matrices, due to (2.5), we obtain a set of shape matrices Y^1, \dots, Y^k with corresponding $\alpha_1, \dots, \alpha_k$.

As a final step, we show that this decomposition yields the matrix A . By our construction, left and right boundary of the intervals are defined according to non-zero elements of matrices L and R respectively. Therefore

$$\begin{aligned} \sum_{k=1}^n \alpha_{mk}^\ell &= \sum_{k:\ell_m^k \leq n} \alpha_k, \\ \sum_{k=1}^n \alpha_{mk}^r &= \sum_{k:r_m^k \leq n} \alpha_k. \end{aligned}$$

On the other hand,

$$\sum_{k:\ell_m^k \leq n < r_m^k} \alpha_k = \sum_{k:\ell_m^k \leq n} \alpha_k - \sum_{k:r_m^k \leq n} \alpha_k = \sum_{k=1}^n \alpha_{mk}^\ell - \sum_{k=1}^n \alpha_{mk}^r.$$

Thus due to (2.6)

$$\sum_{k:\ell_m^k \leq n < r_m^k} \alpha_k = a_{mn},$$

i.e.

$$\sum_{k \in K} \alpha_k Y^k = A.$$

□

Remark 1 *In the proof of Theorem 1 we do not use the integrality of the given matrix and we do not require that matrices L and R be integral. Therefore, our results are true for any matrix with non negative elements and decomposition of it with nonnegative coefficients. Constraint (2.2) is the representation of the intensity cover constraint in terms of matrices L and R . The constraints (2.3) and (2.4) present the leaf collision constraint for opposing leaves in adjacent rows. Note that we do not need any constraint for leaves in same row. The last constraint (2.5) presents the decomposition time; and it is same for all decompositions associated with matrices L and R .*

Now we adopt the theorem for unconstrained decomposition. The proof of the next theorem is the same as the constrained case, since we just omitted the conditions which were related with leaf collision.

Corollary 1 *A has a decomposition with decomposition time $DT(\alpha)$ if and only if there exist $M \times (N + 1)$ matrices $L = (\alpha_{mn}^\ell)$ and $R = (\alpha_{mn}^r)$ with non-negative elements such that*

$$L - R = \tilde{A} \quad (2.8)$$

$$DT(\alpha) = \sum_{k \in \mathcal{K}'} \alpha_k = \sum_{n \in \mathcal{N}} \alpha_{pn}^\ell = \sum_{n \in \mathcal{N}} \alpha_{mn}^r \quad \forall p, m \in \mathcal{M} \quad (2.9)$$

2.2 Invariant Property

Note that \tilde{A} can be written as difference $\tilde{A} = \tilde{L} - \tilde{R}$ of non-negative integer matrices \tilde{L} and \tilde{R} defined as follows: $\tilde{L} := (\tilde{\alpha}_{mn}^\ell)_{\substack{m \in \mathcal{M} \\ n \in \mathcal{N}}}$, $\tilde{R} := (\tilde{\alpha}_{mn}^r)_{\substack{m \in \mathcal{M} \\ n \in \mathcal{N}}}$ with

$$\begin{aligned} \tilde{\alpha}_{mn}^\ell &= \max\{0, a_{mn} - a_{m,n-1}\}, \\ \tilde{\alpha}_{mn}^r &= \max\{0, a_{m,n-1} - a_{mn}\}. \end{aligned} \quad (2.10)$$

Using this notation, for matrices L and R , which satisfy (2.2) – (2.4), we get

$$\tilde{\alpha}_{mn}^\ell - \tilde{\alpha}_{mn}^r = a_{mn} - a_{m,n-1} = \alpha_{mn}^\ell - \alpha_{mn}^r.$$

and

$$\begin{aligned} \alpha_{mn}^\ell &\geq \tilde{\alpha}_{mn}^\ell \\ \alpha_{mn}^r &\geq \tilde{\alpha}_{mn}^r \end{aligned}$$

for all $m \in \mathcal{M}$, $n \in \mathcal{N}$.

Thus we can represent α_{mn}^ℓ and α_{mn}^r in terms of $\tilde{\alpha}_{mn}^\ell$ and $\tilde{\alpha}_{mn}^r$ by using a single variable w_{mn}

$$\begin{aligned}\alpha_{mn}^\ell &= \tilde{\alpha}_{mn}^\ell + w_{mn} \\ \alpha_{mn}^r &= \tilde{\alpha}_{mn}^r + w_{mn}\end{aligned}\tag{2.11}$$

where $w_{mn} \geq 0$.

As we have seen from the proof of Theorem 1 α_{mn}^ℓ and α_{mn}^r present the total time used in the decomposition as a left and a right end of interval, respectively, in row m at position n .

Consequently, $\tilde{\alpha}_{mn}^\ell$ and $\tilde{\alpha}_{mn}^r$, which are calculated from a given matrix directly, present the minimal total time to be used in a decomposition by the right and left ends. To be short we call it necessary time to be used by the left and right and at this position, respectively. Then (2.11) shows an interesting *invariant* property of decomposability that in any decomposition at any position, the time used by left end exceeds the necessary time as much as right end exceeds the necessary time at this position, i.e. for any decomposition

$$\alpha_{mn}^\ell - \alpha_{mn}^r = \tilde{\alpha}_{mn}^\ell - \tilde{\alpha}_{mn}^r = \text{Const}$$

Remark 2 *Invariant property holds for any matrix with non negative elements and decomposition of it into consecutive one's matrices with nonnegative coefficients. Moreover, the following theorem shows that this property holds for decomposition of a matrix into binary matrices with non-negative coefficients.*

Consider any decomposition of a matrix, with non-negative elements, into positive linear composition of binary matrices. As we used before, let us denote by L and R the matrices corresponding to the total time used by left and right end positions in the decomposition.

Theorem 2 *In any decomposition of non-negative matrix A , into positive linear combination of binary matrices, the corresponding matrices $L = (\alpha_{mn}^\ell)$ and $R = (\alpha_{mn}^r)$ satisfy the following*

$$L - R = \tilde{A}\tag{2.12}$$

and for all $m = 1, \dots, M$ and $n = 1, \dots, N + 1$

$$\begin{aligned}\alpha_{mn}^\ell &= \tilde{\alpha}_{mn}^\ell + w_{mn} \\ \alpha_{mn}^r &= \tilde{\alpha}_{mn}^r + w_{mn}\end{aligned}$$

for some $w_{mn} \geq 0$.

Proof: Consider any decomposition

$$A = \sum_{k \in \mathcal{K}'} \alpha_k Y^k$$

Since Y^k is a binary matrix, each row of matrices Y^k may have several intervals, i.e. $[\ell_{m_1}^k, r_{m_1}^k), \dots, [\ell_{m_\zeta}^k, r_{m_\zeta}^k)$. Therefore,

$$\begin{aligned} \alpha_{mn}^\ell &= \sum_{k: \exists p: \ell_{m_p}^k = n} \alpha_k, \\ \alpha_{mn}^r &= \sum_{k: \exists p: r_{m_p}^k = n} \alpha_k. \end{aligned}$$

From the element wise presentation of the decomposition

$$\begin{aligned} a_{mn} &= \sum_{k: \exists p: \ell_{m_p}^k < n < r_{m_p}^k} \alpha_k + \sum_{k: \exists p: \substack{\ell_{m_p}^k = n \\ r_{m_p}^k > n}} \alpha_k, \\ a_{m,n-1} &= \sum_{k: \exists p: \ell_{m_p}^k < n < r_{m_p}^k} \alpha_k + \sum_{k: \exists p: \substack{\ell_{m_p}^k < n \\ r_{m_p}^k = n}} \alpha_k \end{aligned}$$

we get

$$\begin{aligned} a_{mn} - a_{m,n-1} &= \sum_{k: \exists p: \substack{\ell_{m_p}^k = n \\ r_{m_p}^k > n}} \alpha_k - \sum_{k: \exists p: \substack{\ell_{m_p}^k < n \\ r_{m_p}^k = n}} \alpha_k \\ &= \left(\sum_{k: \exists p: \substack{\ell_{m_p}^k = n \\ r_{m_p}^k > n}} \alpha_k + \sum_{k: \exists p: \substack{\ell_{m_p}^k = n \\ r_{m_p}^k = n}} \alpha_k \right) - \left(\sum_{k: \exists p: \substack{\ell_{m_p}^k < n \\ r_{m_p}^k = n}} \alpha_k + \sum_{k: \exists p: \substack{\ell_{m_p}^k = n \\ r_{m_p}^k = n}} \alpha_k \right) \\ &= \sum_{k: \exists p: \ell_{m_p}^k = n} \alpha_k - \sum_{k: \exists p: r_{m_p}^k = n} \alpha_k = \alpha_{mn}^\ell - \alpha_{mn}^r. \end{aligned}$$

Thus L and R satisfy (2.12). Considering the definition of \tilde{A} , (2.10), and doing the same analysis as we did before for constrained case we get (2.13). \square

2.3 Decomposability on a Given Set

Suppose, a set of shape matrices Q be given. Then the next Theorem gives the answer whether there exists a decomposition of the given matrix, with decomposition time DT , using shape matrices from the given set Q .

Theorem 3 *A has a decomposition in Q with decomposition time $DT(\alpha)$ if and only if there exist $M \times (N + 1)$ matrices $L = (\alpha_{mn}^\ell)$ and $R = (\alpha_{mn}^r)$ with non-negative elements such that (2.2)–(2.5) and the following equations satisfied for some non-negative α_q , $q = 1, \dots, |Q|$*

$$\alpha_{mn}^\ell = \sum_{q=1}^{|Q|} \alpha_q \ell_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \quad (2.13)$$

$$\alpha_{mn}^r = \sum_{q=1}^{|Q|} \alpha_q r_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \quad (2.14)$$

where

$$\ell_{mn}^q = \begin{cases} 1 & \text{if in } S_q \text{ left end of interval in row } m \text{ is at position } n \\ 0 & \text{otherwise} \end{cases}$$

and

$$r_{mn}^q = \begin{cases} 1 & \text{if in } S_q \text{ right end of interval in row } m \text{ is at position } n \\ 0 & \text{otherwise} \end{cases}$$

Proof:

- "⇒" Proof follows immediately from Theorem 1. Suppose we have a decomposition, without loss of generality we can present the decomposition in the following way

$$A = \sum_{q=1}^{|Q|} \alpha_q S_q$$

where $\alpha_q = 0$ if S_q is not used in the decomposition. Construct matrices L and R according to (2.13) and (2.14). Then α_{mn}^ℓ and α_{mn}^r present the total time used by right and left interval ends at this position, respectively. Due to the invariant property we get

$$\alpha_{mn}^\ell - \alpha_{mn}^r = \sum_{q=1}^{|Q|} \alpha_q \ell_{mn}^q - \sum_{q=1}^{|Q|} \alpha_q r_{mn}^q = Const = a_{mn} - a_{m,n-1}$$

i.e. conditions (2.2) and (2.5) hold. Since $S_q, \forall q \in Q$, is a shape matrix therefore by definition of shape matrix

$$\ell_m^q \leq r_{m+1}^q \quad \text{and} \quad \ell_{m+1}^q \leq r_m^q$$

which yields (2.3) and (2.4).

- "⇐" Suppose that there exist matrices L and R; and $\alpha_q, q = 1, \dots, |Q|$, which satisfy the conditions of the theorem. From (2.2), as we did before in the proof of Theorem 1, we get for all $n \in \mathcal{N}$

$$\sum_{k=1}^n \alpha_{mk}^\ell - \sum_{k=1}^n \alpha_{mk}^r = a_{mn}$$

due to (2.13) and (2.14) we can write it

$$\begin{aligned} \sum_{k=1}^n \sum_{q=1}^{|Q|} \alpha_q (\ell_{mk}^q - r_{mk}^q) &= a_{mn} \Rightarrow \\ \sum_{k=1}^n \left(\alpha_q \sum_{q=1}^{|Q|} (\ell_{mk}^q - r_{mk}^q) \right) &= a_{mn} \Rightarrow \\ \sum_{q=1}^{|Q|} \left(\alpha_q \sum_{k=1}^n (\ell_{mk}^q - r_{mk}^q) \right) &= a_{mn} \end{aligned}$$

By definition of ℓ_{mn}^q and r_{mn}^q

$$\sum_{k=1}^n (\ell_{mk}^q - r_{mk}^q) = \begin{cases} 0 & \text{if } \ell_m, r_m \leq n \text{ or } \ell_m, r_m > n \\ 1 & \text{if } \ell_m \leq n \text{ and } r_m > n \end{cases}$$

Therefore,

$$\sum_{q: \ell_m^q \leq n < r_m^q} \alpha_q = a_{mn}$$

i.e.

$$\sum_{q=1}^{|Q|} \alpha_q S_q = A$$

□

The version of the theorem for unconstrained case is as follows:

Corollary 2 *A has a decomposition in Q with decomposition time $DT(\alpha)$ if and only if there exist $M \times (N + 1)$ matrices $L = (\alpha_{mn}^\ell)$ and $R = (\alpha_{mn}^r)$ with non-negative elements such that (2.8)–(2.9) and the following equations satisfied for some non-negative α_q , $q = 1, \dots, |Q|$*

$$\alpha_{mn}^\ell = \sum_{q=1}^{|Q|} \alpha_q \ell_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \quad (2.15)$$

$$\alpha_{mn}^r = \sum_{q=1}^{|Q|} \alpha_q r_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \quad (2.16)$$

where

$$\ell_{mn}^q = \begin{cases} 1 & \text{if in } S_q \text{ left end of interval in row } m \text{ is at position } n \\ 0 & \text{otherwise} \end{cases}$$

and

$$r_{mn}^q = \begin{cases} 1 & \text{if in } S_q \text{ right end of interval in row } m \text{ is at position } n \\ 0 & \text{otherwise} \end{cases}$$

Chapter 3

Decomposition Time Problem

3.1 Compact Model For Decomposition Time Problem

According to Theorem 1 solving the decomposition time problem is equivalent to finding one of the pairs of non-negative integer matrices L and R which corresponds to an optimal solution.

According to (2.5) the total decomposition time is, in terms of L and R ,

$$\sum_{k=1}^{N+1} \alpha_{mk}^{\ell} = \sum_{k=1}^{N+1} \tilde{\alpha}_{mk}^{\ell} + \sum_{k=1}^{N+1} w_{mk} = DT_m + \sum_{k=1}^{N+1} w_{mk}$$

where m is the index of any row of A . Therefore, we can use theorem 1 to formulate the decomposition time problem as the following integer linear programming problem (DT-IP)

Let us consider matrices, to simplify the presentation of the matrix corresponding to the constraints (3.4)-(3.6), defined as follows:

$$\begin{aligned}
B_k &:= \{b_{ij}^k\}_{M, N-1} & \text{where} & \quad b_{ij}^k = \begin{cases} -1, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases} \\
\Delta &= \{\delta_{ij}\}_{N-1, N-1} & \text{where} & \quad \delta_{ij} = \begin{cases} 1, & \text{if } i \geq j \\ 0, & \text{otherwise} \end{cases} & \text{lower triangle matrix} \\
-\Delta &= \{\delta_{ij}^-\}_{N-1, N-1} := -1 * \Delta \\
\underline{1} &:= (1, 1, \dots, 1)^T \in R^M
\end{aligned}$$

Then matrix B , corresponding to the constraints (3.4)-(3.6), can be presented as follows:

$$B = \begin{pmatrix} \underline{1} & B_1 & B_2 & B_3 & \dots & B_{M-1} & B_M \\ & \Delta & -\Delta & & & & \\ & & \Delta & -\Delta & & & \\ & & & & \ddots & & \\ & & & & & & \Delta & -\Delta \\ -\Delta & \Delta & & & & & & \\ & -\Delta & \Delta & & & & & \\ & & & & \ddots & & & \\ & & & & & & -\Delta & \Delta \end{pmatrix} \quad (3.7)$$

Where:

- first column presents coefficients corresponding to the variable T ;
- a column corresponding to the B_k present coefficients of the variables w_{kj} , $j = 2, \dots, N$;
- first row corresponds to (3.4)
- next $M - 1$ rows correspond to (3.5)
- last $M - 1$ rows correspond to (3.6)

Theorem 4 (Ghouila-Houri 1962, [45]) *A matrix is a totally unimodular if and only if each subset $J \subseteq [n]$ of the columns can be partitioned into two classes J_+ and J_- such that for each row $i \in [m]$ we have*

$$\left| \sum_{j \in J_+} b_{ij} - \sum_{j \in J_-} b_{ij} \right| \leq 1. \quad (3.8)$$

Corollary 3 *Matrix \mathbf{B} , defined as in (3.7), is a totally unimodular matrix.*

Proof: Let J be the set of column indexes of \mathbf{B} .

Let us denote by J_k the set of column indexes of B corresponding to the matrix B_k , i.e.

$$J_k = \{ (k-1) * (N-1) + 2, (k-1) * (N-1) + 3, \dots, (k-1) * (N-1) + N \}$$

Then

$$J = \bigcup_{k=0}^M J_k = \{ 1, 2, \dots, 1 + M * (N-1) \}$$

where

$$J_0 := \{1\}.$$

Consider an arbitrary subset J' of J . Then

$$J' = \bigcup_{k=0}^M J'_k$$

where J'_k is either $J'_k \subseteq J_k$ or $J'_k = \emptyset$, for all $k = 0, 1, \dots, M$.

For each nonempty subset $J'_k = \{i_1, i_2, \dots, i_{p_k}\}$, $0 \leq k \leq M$, we numerate indexes such that $i_1 \leq i_2 \leq \dots \leq i_{p_k}$, and denote the set of indexes with odd numbers by J'_k and with even numbers by J''_k , i.e.

$$\begin{aligned} i_p \in J'_k & \text{ if } p \equiv 1(\text{mod } 2) \\ i_p \in J''_k & \text{ if } p \equiv 0(\text{mod } 2) \\ 1 \leq p \leq p_k \end{aligned}$$

Then from our construction of the sets it is clear that

$$\text{either } |J'_k| = |J''_k| \text{ or } |J'_k| = |J''_k| + 1 \quad (3.9)$$

and

$$J'_k = J'_k \cup J''_k$$

Therefore, we get a partitioning of J' into two sets:

$$\begin{aligned} J_+ & := \bigcup_{k=0}^M J'_k \\ J_- & := \bigcup_{k=0}^M J''_k \end{aligned}$$

We complete the proof by showing that for our partitioning the condition (3.8) is valid for all rows of B . Consider an arbitrary row i of B .

- *Case 1.* If i is one of the rows which corresponds to constraints (3.4) then from the special structure of the matrices B_k we get

$$\sum_{j \in J_+} b_{ij} - \sum_{j \in J_-} b_{ij} = \begin{cases} 1 + \sum_{j \in J_+^i} b_{ij} - \sum_{j \in J_-^i} b_{ij} = 1 - |J_+^i| + |J_-^i| & \text{if } 1 \in J^i \\ \sum_{j \in J_+^i} b_{ij} - \sum_{j \in J_-^i} b_{ij} = -|J_+^i| + |J_-^i| & \text{otherwise} \end{cases}$$

From (3.9) follows that row i satisfies (3.8).

- *Case 2.* i is one of the rows which correspond to constraints (3.5) then from the special structure of the matrix B there exist k such that

$$b_{ij} = 0 \quad \text{if} \quad j \notin J_k \quad \text{and} \quad j \notin J_{k+1}$$

Therefore,

$$\begin{aligned} \left| \sum_{j \in J_+} b_{ij} - \sum_{j \in J_-} b_{ij} \right| &= \\ &= \left| \sum_{j \in J_+^k} \delta_{kj} - \sum_{j \in J_-^k} \delta_{kj} + \sum_{j \in J_+^{k+1}} \delta_{k+1,j} - \sum_{j \in J_-^{k+1}} \delta_{k+1,j} \right| = \\ &= \underbrace{||J_+^k| - |J_-^k|}_{0 \leq, \leq 1} + \underbrace{(-|J_+^{k+1}| + |J_-^{k+1}|)}_{-1 \leq, \leq 0} \leq 1 \end{aligned}$$

- *Case 3.* i is one of the rows which correspond to constraints (3.6).
Proof similar with Case 2.

For an arbitrary row i holds (3.8), therefore by theorem 4 B is TU. \square

Therefore, by solving the linear relaxation of (DT-IP) we get always an integer optimal solution.

Corollary 4 *If A is a binary matrix then (BOT) is equivalent to (NS).*

In other words, for a given binary intensity matrix we can solve at once both problems (BOT) and (NS) in a polynomial time.

3.3 Algorithm and Numerical results

Finally, as a result of our work, we get the following algorithm.

Algorithm 1. (Constrained DT Problem)

Input: Intensity matrix A

Step 0: Compute matrices \tilde{L} and \tilde{R}

Step 1: Solve LP relaxation of (DT-IP)

Step 2: Extract shape matrices with corresponding decomposition coefficients as in the proof of Theorem 1: starting from the left-most non-zero elements in the rows of matrices L and R

Output: Shape matrices and corresponding beam-on times. From now on we will refer to the extraction scheme used in step 2, as a "*leftmost extraction rule*"

The algorithm has been implemented using CPLEX 9.0 embedded in C++, and tested on over 1000 randomly generated intensity maps, for each row of the Table 3.1, with intensity levels from 0 to 15. The average CPU time spent on the algorithm, on AMD Athlon(tm) 64 Processor 2800, 1.81 GHz, 512 MB RAM, is shown in Fig. 3.1

Matrix	CPU Time sec.
10x10	0.0*
15x15	0.0
20x20	0.0
30x30	0.2
40x40	0.4
50x50	0.7
100x100	5.5

Figure 3.1: Average CPU time(sec) for MxN matrix. 0.0 time means that the computation time was less then 0.5 sec

3.4 Nested sequence of Multicriteria Optimization problems

In the following part of the thesis we show that these integer programming problems can be solved by a combinatorial algorithm in polynomial time.

The feasible solutions of (DT-IP) have the following property which will be essential in the development of an efficient algorithm.

Lemma 1 *Let $W = (w_{mn})$ be a feasible solution of (DT-IP). If for any column p , $w_p = (w_{1p}, w_{2p}, \dots, w_{Mp})^T$, there exists $\bar{w} = (\bar{w}_1, \dots, \bar{w}_M)^T \geq 0$ such that $w_p \geq \bar{w}$ and*

$$\begin{aligned} \sum_{k=1}^p \tilde{\alpha}_{m-1,k}^\ell + \sum_{k=1}^{p-1} w_{m-1,k} + \bar{w}_{m-1} &\geq \sum_{k=1}^p \tilde{\alpha}_{mk}^r + \sum_{k=1}^{p-1} w_{mk} + \bar{w}_m, \\ \sum_{k=1}^p \tilde{\alpha}_{mk}^\ell + \sum_{k=1}^{p-1} w_{mk} + \bar{w}_m &\geq \sum_{k=1}^p \tilde{\alpha}_{m-1,k}^r + \sum_{k=1}^{p-1} w_{m-1,k} + \bar{w}_{m-1} \end{aligned}$$

for all $m = 2, \dots, M$ then replacing columns w_p and w_{p+1} of W by \bar{w} and $w_{p+1} + w_p - \bar{w}$, respectively, we get a feasible solution of (DT-IP) with the same objective value as W .

Proof: The sum of the columns (vectors) \bar{w} and $w_{p+1} + w_p - \bar{w}$ is the same as it was before, $w_{p+1} + w_p$. Therefore, this replacement does not change the objective function value and it may only affect the constraints of (DT-IP) corresponding to $n = p$. By the given condition on \bar{w} these are satisfied. \square

Based on Lemma 1, we solve (DT-IP) recursively by solving a sequence of multiobjective integer programs (SP_n), $n = 1, \dots, N+1$, in which the input data is defined by the output of (SP_k), $k < n$. (SP_n) is as follows.

$$\min \begin{pmatrix} w_{1n} \\ w_{2n} \\ \vdots \\ w_{Mn} \end{pmatrix}$$

$$\text{s.t. } DTL_{m-1}^n + w_{m-1,n} \geq DTR_m^n + w_{mn} \quad \forall m \in \mathcal{M} \setminus \{1\} \quad (3.10)$$

$$DTL_m^n + w_{mn} \geq DTR_{m-1}^n + w_{m-1,n} \quad \forall m \in \mathcal{M} \setminus \{1\} \quad (3.11)$$

$$w_{mn} \geq 0 \quad \forall m \in \mathcal{M}.$$

Here

$$\begin{aligned} DTL_m^n &= \sum_{k=1}^n \tilde{\alpha}_{mk}^\ell + \sum_{k=1}^{n-1} w_{mk}^*, \\ DTR_m^n &= \sum_{k=1}^n \tilde{\alpha}_{mk}^r + \sum_{k=1}^{n-1} w_{mk}^* \end{aligned}$$

where $(w_{1k}^*, w_{2k}^*, \dots, w_{mk}^*)^T$ is the optimal solution of (SP_k) , $k < n$.

Due to (2.7) and (2.11) we get

$$DTL_m^n \geq DTR_m^n \quad \forall m \in \mathcal{M} \quad (3.12)$$

a property which we will use later on.

The next result shows that (SP_n) is, indeed, well posed and that (SP_n) , $n \in \mathcal{N}$, yields an optimal solution of (DT-IP).

Proposition 1 (SP_n) has a unique Pareto optimal solution.

Proof: We show the result by contradiction. Assume that there exist two different Pareto optimal solutions $\bar{w} = (\bar{w}_{1n}, \dots, \bar{w}_{Mn})$ and $\hat{w} = (\hat{w}_{1n}, \dots, \hat{w}_{Mn})$ to (SP_n) . Consider $w = (w_{1n}, w_{2n}, \dots, w_{Mn})$ defined by

$$w_{mn} := \min\{\bar{w}_{mn}, \hat{w}_{mn}\},$$

i.e., $w \not\leq \bar{w}$ and $w \not\leq \hat{w}$.

Consider the constraints of (SP_n) corresponding to an arbitrary m

$$\begin{aligned} DTL_{m-1}^n + w_{m-1,n} &\geq DTR_m^n + w_{mn}, \\ DTL_m^n + w_{mn} &\geq DTR_{m-1}^n + w_{m-1,n}. \end{aligned}$$

If

$$\begin{aligned} w_{m-1,n} &= \bar{w}_{m-1,n}, \\ w_{mn} &= \bar{w}_{mn} \end{aligned}$$

or

$$\begin{aligned} w_{m-1,n} &= \hat{w}_{m-1,n}, \\ w_{mn} &= \hat{w}_{mn} \end{aligned}$$

then the inequalities hold since \bar{w} and \hat{w} are feasible solutions.

If $w_{m-1,n} = \bar{w}_{m-1,n}$ and $w_{mn} = \hat{w}_{mn}$, then from

$$\begin{aligned} \bar{w}_{m-1,n} &\leq \hat{w}_{m-1,n}, \\ \bar{w}_{mn} &\geq \hat{w}_{mn} \end{aligned}$$

it follows that

$$\begin{aligned} DTL_{m-1}^n + \bar{w}_{m-1,n} &\geq DTR_m^n + \bar{w}_{mn} &&\geq DTR_m^n + \hat{w}_{mn}, \\ DTL_m^n + \hat{w}_{mn} &\geq DTR_{m-1}^n + \hat{w}_{m-1,n} &&\geq DTR_{m-1}^n + \bar{w}_{m-1,n}, \end{aligned}$$

i.e., w is a feasible solution of (SP_n) and because $w \not\leq \bar{w}$, $w \not\leq \hat{w}$ that contradicts that \bar{w} , \hat{w} are Pareto optimal solutions. \square

Algorithm 3.4.1 (Minimum C1 Decomposition Time into Shape Matrices)

Input: Matrix A

Output: Decomposition of A into shape matrices with $\min DT(\alpha)$

(1.) Compute $\tilde{\alpha}_{mn}^\ell, \tilde{\alpha}_{mn}^r, \forall m, n$.

(2.) For $n = 1$ to $N + 1$

Solve (SP_n) (with Algorithm 3.4.2)

(3.) Compute matrices L and R ; and $DT(\alpha)$

(4.) Set $k := 0$

(5.) While $DT(\alpha) \neq 0$ do

Consider leftmost non-zero elements

$\alpha_{m\ell_m}^\ell$ and $\alpha_{mr_m}^r, m = 1, \dots, M$, in each row of L and R

$k := k + 1$

Extract shape matrix

$Y^k = Y([\ell_m, r_m])_{m \in \mathcal{M}}$ with

$\alpha_k = \min\{\alpha_{1\ell_1}^\ell, \dots, \alpha_{M\ell_M}^\ell, \alpha_{1r_1}^r, \dots, \alpha_{Mr_M}^r\}$

Set $DT(\alpha) := DT(\alpha) - \alpha_k$

Update L and R .

end while

It remains to show how to solve (SP_n) , $n \in \mathcal{N}$. This can be done by the following combinatorial algorithm.

Algorithm 3.4.2 (Solving (SP_n))

Input: $DTL_m^n, DTR_m^n, \quad \forall m = 1, \dots, M.$

Output: $w_{mn}^*, \quad \forall m = 1, \dots, M$

(1.) $w_{mn} := 0, \quad \forall m = 1, \dots, M$

(2.) For $m = 2$ to M do

if $DTL_m^n + w_{mn} \leq DTR_{m-1}^n + w_{m-1,n}$
then $w_{mn} := DTR_{m-1}^n - DTL_m^n + w_{m-1,n}$
else $A(m)$

end for

Function $A(p)$

if $DTL_{p-1}^n + w_{p-1,n} < DTR_p^n + w_{p,n}$
then $w_{p-1,n} := DTR_p^n - DTL_{p-1}^n + w_{p,n}$

if $p \geq 3$

then $p := p - 1$

$A(p)$

end if

end if

end Function

Remark 3 We do not require w_{mn} , for all $m = 1, \dots, M$, $n = 1, \dots, N + 1$, to be integers. But if the given matrix A is an integer matrix then Algorithm 3.4.2 provides an integer solution.

Theorem 5 Algorithm 3.4.2 finds the optimal solution of (SP_n) in $\mathcal{O}(M^2)$ time.

Algorithm 3.4.1 solves $(DT-IP)$ in $\mathcal{O}(NM^2)$ time.

Proof: Obviously, the time complexity of Algorithm 3.4.2 is $\mathcal{O}(M^2)$. If we can prove that Algorithm 3.4.2 solves (SP_n) to optimality, the time complexity of Algorithm 3.4.1 is $\mathcal{O}(NM^2)$. Hence the validity of Algorithm 3.4.2 remains to be shown. We do it by induction.

$m = 1$: We do not have any constraints, except $w_{1n} \geq 0$. Therefore, the initialization $w_{1n} = 0$ is the optimal solution.

$m = 2$: In this case we have just two constraints

$$DTL_1^n + w_{1n} \geq DTR_2^n + w_{2n}, \quad (3.13)$$

$$DTL_2^n + w_{2n} \geq DTR_1^n + w_{1n} \quad (3.14)$$

and by initialization $w_{1n} = w_{2n} = 0$ is the lower bound on the values of w_{1n} and w_{2n} . We will tighten these lower bounds next to obtain a feasible solution for (SP_n) which is thus optimal.

- Case 1: $DTL_2^n \leq DTR_1^n$. Then $w_{2n} = DTR_1^n - DTL_2^n$ from step (2.) is a lower bound by (3.13) and (3.14); and satisfies $DTR_2^n + w_{2n} = DTR_2^n + DTR_1^n - DTL_2^n \leq DTR_2^n + DTR_1^n - DTR_2^n = DTR_1^n \leq DTL_1^n$, due to (3.12). Hence $(w_{1n} = 0, w_{2n})$ is feasible.
- Case 2: $DTL_2^n > DTR_1^n$. Then $w_{2n} = 0$ is the lower bound due to (3.13) and (3.14). The lower bound for w_{2n} is tightened using Function A(2)
 - If $DTL_1^n < DTR_2^n$ then using (3.12) $w_{1n} = DTR_2^n - DTL_1^n$ satisfies $DTR_1^n + w_{1n} = DTR_1^n + DTR_2^n - DTL_1^n \leq DTR_1^n + DTR_2^n - DTR_1^n = DTR_2^n \leq DTL_2^n$ such that $(w_{1n}, w_{2n} = 0)$ is feasible.
 - If $DTL_1^n \geq DTR_2^n$ then $(w_{1n} = 0, w_{2n} = 0)$ is feasible.

$m < M$: Assume that Algorithm 3.4.2 yields the optimal solution of (SP_n) for all $m < M$.

$m = M$: Running the algorithm until $m = M - 1$, in the loop (2.), we get by the induction hypothesis the optimal solution to (SP_n) defined for rows $1, \dots, M - 1$. This solution can serve as a lower bound for w_{mn} , $m = 1, \dots, M - 1$ of problem (SP_n) defined for rows $1, \dots, M$. Now we tighten this bound with respect to constraints

$$DTL_{M-1}^n + w_{M-1,n} \geq DTR_M^n + w_{Mn}, \quad (3.15)$$

$$DTL_M^n + w_{Mn} \geq DTR_{M-1}^n + w_{M-1,n} \quad (3.16)$$

which contain variable w_{Mn} .

- Case 1: If $DTL_M^n \leq DTR_{M-1}^n + w_{M-1,n}$ then the lower bound for w_{Mn} is $w_{Mn} = DTR_{M-1}^n + w_{M-1,n} - DTL_M^n$, which satisfies both inequalities since $DTR_M^n + DTR_{M-1}^n + w_{M-1,n} - DTL_M^n \leq DTR_M^n + DTR_{M-1}^n + w_{M-1,n} - DTR_M^n = DTR_{M-1}^n + w_{M-1,n} \leq DTL_{M-1}^n + w_{M-1,n}$ due to (3.12).
- Case 2: If $DTL_M^n > DTR_{M-1}^n + w_{M-1,n}$ then $w_{Mn} = 0$
 - If the (3.15) is satisfied then the algorithm terminates
 - Otherwise, i.e., if

$$DTL_{M-1}^n + w_{M-1,n} < DTR_M^n$$

then we increase(tighten) the lower bound for $w_{M-1,n}$ found in the previous step:

$$w_{M-1,n} := DTR_M^n - DTL_{M-1}^n.$$

The increase of value $w_{M-1,n}$ can affect only two constraints for $m = M$ and $m = M - 1$ where $w_{M-1,n}$ is on the right hand side. The first of these inequalities, namely (3.16), holds by the choice of $w_{M-1,n}$ and (3.12). The second one is

$$DTL_{M-2}^n + w_{M-2,n} \geq DTR_{M-1}^n + w_{M-1,n}.$$

If this holds an optimal solution is obtained. Otherwise, the algorithm updates the lower bound for $w_{M-2,n}$ and checks the inequalities where $w_{M-2,n}$ is on the right hand side.

The algorithm iterates the above procedure until all updated lower bounds are feasible for (SP_n) . Thus we have the optimal solution.

□

Example 3.4.1

$$A = \begin{pmatrix} 5 & 10 & 6 \\ 4 & 1 & 1 \\ 7 & 0 & 0 \end{pmatrix}.$$

The corresponding matrices

$$\tilde{L} = \begin{pmatrix} 5 & 5 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{R} = \begin{pmatrix} 0 & 0 & 4 & 6 \\ 0 & 3 & 0 & 1 \\ 0 & 7 & 0 & 0 \end{pmatrix}$$

of A are defined according to (2.10). Solving iteratively subproblems (SP_n) we get the optimal solution to $(DT-IP)$. In the table below we show the input data and solutions of subproblems (SP_n)

$n = 1$:	Input:	$DTL_1^1 = 5,$	$DTL_2^1 = 4,$	$DTL_3^1 = 7,$
		$DTR_1^1 = 0,$	$DTR_2^1 = 0,$	$DTR_3^1 = 0$
	Output:	$w_{m1}^* = 0,$	$m = 1, 2, 3$	
$n = 2$:	Input:	$DTL_1^2 = 10,$	$DTL_2^2 = 4,$	$DTL_3^2 = 7,$
		$DTR_1^2 = 0,$	$DTR_2^2 = 3,$	$DTR_3^2 = 7$
	Output:	$w_{12}^* = 0,$	$w_{32}^* = 0, w_{22}^* = 3$	
$n = 3$:	Input:	$DTL_1^3 = 10,$	$DTL_2^3 = 7,$	$DTL_3^3 = 7,$
		$DTR_1^3 = 4,$	$DTR_2^3 = 6,$	$DTR_3^3 = 7$
	Output:	$w_{m3}^* = 0,$	$m = 1, 2, 3$	
$n = 4$:	Input:	$DTL_1^4 = 10,$	$DTL_2^4 = 7,$	$DTL_3^4 = 7,$
		$DTR_1^4 = 10,$	$DTR_2^4 = 7,$	$DTR_3^4 = 7$
	Output:	$w_{14}^* = 0,$	$w_{24}^* = 3,$	$w_{34}^* = 3$

For $n = 2$ we have $DTL_2^2 = 4$ and $DTR_3^2 = 7$, therefore w_{22}^* is set to 3. No further changes to w are necessary. For $n = 4$ $DTL_2^4 = 7 > DTR_1^4 = 10$ thus $w_{24} \geq 3$. This results in $DTR_2^4 + w_{24} = 10 > DTL_3^4 = 7$ and $w_{34}^* = 3$. Since all inequalities are satisfied, $w_{24}^* = 3$, too.

Using the solution of $(DT-IP)$

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

we compute the matrices

$$L = \begin{pmatrix} 5 & 5 & 0 & 0 \\ 4 & 3 & 0 & 3 \\ 7 & 0 & 0 & 3 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 0 & 4 & 6 \\ 0 & 6 & 0 & 4 \\ 0 & 7 & 0 & 3 \end{pmatrix}$$

which correspond to an optimal solution of the decomposition time problem with $DT(\alpha) = 10$. Extracting shape matrices, with respect to the leftmost non-zero elements of L and R , we get the following decomposition

$$\begin{aligned} Y^1 &= Y \begin{pmatrix} [1, 3] \\ [1, 2] \\ [1, 2] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_1 = 4 \\ Y^2 &= Y \begin{pmatrix} [1, 4] \\ [2, 2] \\ [1, 2] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_2 = 1 \\ Y^3 &= Y \begin{pmatrix} [2, 4] \\ [2, 2] \\ [1, 2] \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_3 = 1 \\ Y^4 &= Y \begin{pmatrix} [2, 4] \\ [2, 4] \\ [1, 2] \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_4 = 1 \\ Y^5 &= Y \begin{pmatrix} [2, 4] \\ [4, 4] \\ [4, 4] \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \alpha_5 = 3 \end{aligned}$$

with $K_A = 5$ and $\sum_{k=1}^{K_A} \alpha_k = 10 = DT$.

3.5 Upper Bound on End Positions

Here we consider upper bounds on total times for end positions within the given decomposition time. Assume that we are looking for a decomposition which has

exactly DT total decomposition time. We may assume that $DT \geq DT^*$, otherwise there is no such a decomposition exists. We can use Algorithm 3.4.1 to solve this problem. But we need to add an additional constraint– total decomposition time equals to DT – into the last subproblem (SP_{N+1}). Let L and R be matrices obtained in this way.

Proposition 2 *Last columns of matrices L and R , $\alpha_{m,N+1}^\ell$ and $\alpha_{m,N+1}^r$, $m = 1, \dots, M$, are upper bound on the total time to be used, in any decomposition with decomposition time DT , as a left and right end of intervals at position $N + 1$, respectively.*

Proof: Proof follows immediately from uniqueness of Pareto optimal solutions of problems (SP_n), $n = 1, \dots, N + 1$. \square

Suppose we have computed matrices L and R , which solve the sequence of multicriteria optimization problems (SP_n), $n = 1, \dots, N + 1$. Now consider a sequence of multicriteria optimization problems (USP_n), $n = N, \dots, 1$, defined as follows

$$\begin{aligned} & \max \begin{pmatrix} w_{1n} \\ w_{2n} \\ \vdots \\ w_{Mn} \end{pmatrix} \\ \text{s.t. } & UDTL_{m-1}^n + w_{m-1,n} \geq UDTR_m^n + w_{mn} \quad \forall m \in \mathcal{M} \setminus \{1\} \quad (3.17) \\ & UDTL_m^n + w_{mn} \geq UDTR_{m-1}^n + w_{m-1,n} \quad \forall m \in \mathcal{M} \setminus \{1\} \quad (3.18) \\ & \min\{\alpha_{m,n+1}^\ell, \alpha_{m,n+1}^r\} \geq w_{mn} \geq 0 \quad \forall m \in \mathcal{M}. \quad (3.19) \end{aligned}$$

Here

$$\begin{aligned} UDTL_m^n &= \sum_{k=1}^n \alpha_{mk}^\ell \\ UDTR_m^n &= \sum_{k=1}^n \alpha_{mk}^r \end{aligned}$$

Note that from properties of matrices L and R follow immediately that

$$UDTL_m^n \geq UDTR_m^n \quad (3.20)$$

Each time we update columns $n + 1$ of matrices L and R as follows

$$\begin{aligned} \alpha_{mn}^\ell &= \alpha_{mn}^\ell + w_{mn}^* \\ \alpha_{mn}^r &= \alpha_{mn}^r + w_{mn}^* \end{aligned} \quad (3.21)$$

where w_{mn}^* , $m = 1, \dots, M$, is the optimal solution to (USP_n).

Proposition 3 (USP_n) has a unique Pareto optimal solution.

Proof: We show the result by contradiction. Assume that there exist two different Pareto optimal solutions $\bar{w} = (\bar{w}_{1n}, \dots, \bar{w}_{Mn})$ and $\hat{w} = (\hat{w}_{1n}, \dots, \hat{w}_{Mn})$ to (USP_n). Consider $w = (w_{1n}, w_{2n}, \dots, w_{Mn})$ defined by

$$w_{mn} := \max\{\bar{w}_{mn}, \hat{w}_{mn}\},$$

i.e., $w \succeq \bar{w}$ and $w \succeq \hat{w}$.

Consider the constraints of (USP_n) corresponding to an arbitrary m

$$\begin{aligned} UDTL_{m-1}^n + w_{m-1,n} &\geq UDTR_m^n + w_{mn}, \\ UDTL_m^n + w_{mn} &\geq UDTR_{m-1}^n + w_{m-1,n}. \end{aligned}$$

If

$$\begin{aligned} w_{m-1,n} &= \bar{w}_{m-1,n}, \\ w_{mn} &= \bar{w}_{mn} \end{aligned}$$

or

$$\begin{aligned} w_{m-1,n} &= \hat{w}_{m-1,n}, \\ w_{mn} &= \hat{w}_{mn} \end{aligned}$$

then the inequalities hold since \bar{w} and \hat{w} are feasible solutions.

If $w_{m-1,n} = \bar{w}_{m-1,n}$ and $w_{mn} = \hat{w}_{mn}$, then from

$$\begin{aligned} \bar{w}_{m-1,n} &\geq \hat{w}_{m-1,n}, \\ \bar{w}_{mn} &\leq \hat{w}_{mn} \end{aligned}$$

it follows that

$$\begin{aligned} UDTL_{m-1}^n + \bar{w}_{m-1,n} &\geq UDTL_{m-1}^n + \hat{w}_{m-1,n} \geq UDTR_m^n + \hat{w}_{mn}, \\ UDTL_m^n + \hat{w}_{mn} &\geq UDTL_m^n + \bar{w}_{mn} \geq UDTR_{m-1}^n + \bar{w}_{m-1,n}, \end{aligned}$$

i.e., w is a feasible solution of (SP_n) and because $w \succeq \bar{w}$, $w \succeq \hat{w}$ that contradicts that \bar{w} , \hat{w} are Pareto optimal solutions. \square

Algorithm 3.5.1 (Solving (USP_n))

Input: $UDTL_m^n, UDTR_m^n, \quad \forall m = 1, \dots, M.$

Output: $w_{mn}^*, \quad \forall m = 1, \dots, M$

(1.) $w_{mn} := \min\{\alpha_{m,n+1}^\ell, \alpha_{m,n+1}^r\}, \quad \forall m = 1, \dots, M$

(2.) For $m = 2$ to M do

if $UDTL_{m-1}^n + w_{m-1,n} \leq UDTR_m^n + w_{mn}$

then $w_{mn} := UDTL_{m-1}^n - UDTR_m^n + w_{m-1,n}$

else $A(m)$

end for

Function $A(p)$

if $UDTL_p^n + w_{pn} < UDTR_{p-1}^n + w_{p-1,n}$

then $w_{p-1,n} := UDTL_p^n - UDTR_{p-1}^n + w_{pn}$

if $p \geq 3$

then $p := p - 1$

$A(p)$

end if

end if

end Function

Theorem 6 Algorithm 3.5.1 finds the optimal solution of (USP_n) in $\mathcal{O}(M^2)$ time.

Proof: Obviously, the time complexity of Algorithm 3.5.1 is $\mathcal{O}(M^2)$, if we can prove that Algorithm 3.5.1 solves (USP_n) to optimality. We do it by induction.

$m = 1$: We do not have any constraints, except $\min\{\alpha_{1,n+1}^\ell, \alpha_{1,n+1}^r\} \geq w_{1n}$. Therefore, the initialization $\min\{\alpha_{1,n+1}^\ell, \alpha_{1,n+1}^r\} = w_{1n}$ is the optimal solution.

$m = 2$: In this case we have just two constraints

$$UDTL_1^n + w_{1n} \geq UDTR_2^n + w_{2n}, \quad (3.22)$$

$$UDTL_2^n + w_{2n} \geq UDTR_1^n + w_{1n} \quad (3.23)$$

and by initialization $w_{1n} = \min\{\alpha_{1,n+1}^\ell, \alpha_{1,n+1}^r\}$, $w_{2n} = \min\{\alpha_{2,n+1}^\ell, \alpha_{2,n+1}^r\}$ are upper bounds on the values of w_{1n} and w_{2n} . We will tighten these bounds next to obtain a feasible solution for (USP_n) which is thus optimal.

- Case 1: $UDTL_1^n + w_{1n} \leq UDTR_2^n + w_{2n}$. Then $w_{2n} = UDTL_1^n - UDTR_2^n + w_{1n}$ from step (2.) is a upper bound by (3.22) and (3.23); and moreover satisfies (3.23)

$$\begin{aligned} UDTL_2^n + w_{2n} &= UDTL_2^n + UDTL_1^n - UDTR_2^n + w_{1n} \\ &\geq UDTL_1^n + w_{1n} \geq UDTR_1^n + w_{1n}, \end{aligned}$$

due to (3.20). Hence (w_{1n}, w_{2n}) is feasible.

- Case 2: $UDTL_1^n + w_{1n} > UDTR_2^n + w_{2n}$. Then w_{2n} is the upper bound due to (3.22) and (3.23). The upper bound for w_{1n} is tightened using Function A(2)

- If $UDTL_p^n + w_{pn} < UDTR_{p-1}^n + w_{p-1,n}$ then $w_{1n} := UDTL_2^n - UDTR_1^n + w_{2n}$ is an upper bound for w_{1n} and due to (3.20) it satisfies (3.22)

$$\begin{aligned} UDTL_1^n + w_{1n} &= UDTL_1^n + UDTL_2^n - UDTR_1^n + w_{2n} \\ &\geq UDTL_2^n + w_{2n} \geq UDTR_2^n + w_{2n}, \end{aligned}$$

such that (w_{1n}, w_{2n}) is feasible.

- If $UDTL_p^n + w_{pn} \geq UDTR_{p-1}^n + w_{p-1,n}$ then (w_{1n}, w_{2n}) is feasible.

$m < M$: Assume that Algorithm 3.5.1 yields the optimal solution of (USP_n) for all $m < M$.

$m = M$: Running the algorithm until $m = M - 1$, in the loop (2.), we get by the induction hypothesis the optimal solution to (USP_n) defined for rows $1, \dots, M - 1$. This solution can serve as a lower bound for w_{mn} , $m = 1, \dots, M - 1$ of problem (USP_n) defined for rows $1, \dots, M$. Now we tighten this bound with respect to constraints

$$UDTL_{M-1}^n + w_{M-1,n} \geq UDTR_M^n + w_{Mn}, \quad (3.24)$$

$$UDTL_M^n + w_{Mn} \geq UDTR_{M-1}^n + w_{M-1,n} \quad (3.25)$$

which contain variable w_{Mn} . Initialise $w_{Mn} := \min\{\alpha_{M,n+1}^\ell, \alpha_{M,n+1}^r\}$.

- Case 1: If $UDTL_{M-1}^n + w_{M-1,n} \leq UDTR_M^n + w_{Mn}$ then the upper bound for w_{Mn} is $w_{Mn} = UDTL_{M-1}^n - UDTR_M^n + w_{M-1,n}$, which satisfies both inequalities since

$$\begin{aligned} UDTL_M^n + w_{Mn} &= UDTL_M^n + UDTL_{M-1}^n - UDTR_M^n + w_{M-1,n} \\ &\geq UDTL_{M-1}^n + w_{M-1,n} \geq UDTR_{M-1}^n + w_{M-1,n}, \end{aligned}$$

due to (3.20).

- Case 2: If $UDTL_{M-1}^n + w_{M-1,n} > UDTR_M^n + w_{Mn}$ then
 - If the (3.24) is satisfied then the algorithm terminates
 - Otherwise, i.e., if

$$UDTL_M^n + w_{Mn} < UDTR_{M-1}^n + w_{M-1,n}$$

then we decrease(tighten) the upper bound for $w_{M-1,n}$ found in the previous step:

$$w_{M-1,n} := UDTL_M^n - UDTR_{M-1}^n + w_{Mn}$$

The decrease of value $w_{M-1,n}$ can affect only two constraints for $m = M$ and $m = M - 1$ where $w_{M-1,n}$ is on the left hand side. The first of these inequalities, namely (3.24), holds by the choice of $w_{M-1,n}$ and (3.20). The second one is

$$UDTL_{M-1}^n + w_{M-1,n} \geq UDTR_{M-2}^n + w_{M-2,n}.$$

If this holds an optimal solution is obtained. Otherwise, the algorithm updates the upper bound for $w_{M-2,n}$ and checks the inequalities where $w_{M-2,n}$ is on the left hand side.

The algorithm iterates the above procedure until all updated lower bounds are feasible for (USP_n) . Thus we have the optimal solution.

□

Theorem 7 *Total time defined by (3.21) by solving the sequence of problems (USP_n) is the upper bound for end positions within decomposition time DT .*

Proof:

Suppose that it is not true for some decomposition with decomposition time DT . Consider a column, say k , for which the proposition does not hold. Let \hat{L} and \hat{R} be matrices associated to this decomposition. Due to Lemma 1 we can use Algorithm 3.4.2 for this decomposition for columns $n = 1, \dots, k - 1$.

Then using Algorithm 3.5.1 for columns $n = N, \dots, k$ we get new matrices \tilde{L} and \tilde{R} , where elements in columns k of those matrices are greater or equal to the corresponding elements in matrices \hat{L} and \hat{R} . On the other hand, by our constructive way of algorithms we will have exactly the same columns, except columns k , as in the corresponding matrices L and R , obtained using to the problem Algorithm 3.4.1 and then Algorithm 3.5.1 for columns $n = N, \dots, k$ within the given decomposition time DT. Then by our assumption at least one of the columns k of matrices \tilde{L} and \tilde{R} must have at least one element which is greater than the corresponding element in the corresponding matrix L or R . It is contradicting that multicriteria optimization problems (USP_n) has a unique Pareto optimal solution. \square

Therefore, we can use the following algorithm to compute the upper bounds on end positions within given decomposition time DT.

Algorithm 3.5.2 (Upper Bound On End Positions)

Input: A, DT

Output: *matrices L and R: Upper Bound for each end positions*

(1.) *Compute matrices L and R using Algorithm 3.4.1*

(2.) *For ($n = N + 1; n > 0; n - -$)*

Use Algorithm 3.5.1 for column n

Update column n of matrices L and R

endFor.

Algorithm 3.5.2 finds the upper bounds on end positions in $\mathcal{O}(M^2 N)$ time.

3.6 Unconstrained Decomposition Time Problem

For the unconstrained case due to Theorem 1 and due to the invariant property of decomposition we get the following IP problem

$$\begin{aligned}
 \min \quad & DT(\alpha) \\
 \text{s.t.} \quad & DT_m + \sum_{k=1}^{N+1} w_{mk} = DT(\alpha) \quad \forall m \in \mathcal{M} \\
 & w_{mn} \geq 0, \text{ integer} \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}.
 \end{aligned}$$

Obviously, we can drop the integrality condition since the matrix of coefficients of this problem is submatrix of the coefficient matrix, which is totally unimodular, of the constrained problem, which is totally unimodular.

It is clear that the minimal decomposition time DT^* is defined by the maximal decomposition time of the rows, i.e.

$$DT^* = \max_{m \in \mathcal{M}} \left\{ \sum_{n=1}^{N+1} \tilde{\alpha}_{mn}^\ell \right\} \quad (3.26)$$

Thus, we compute the minimal decomposition time and row of the given matrix A using (3.26). Then, we can compute the sum of w_{mn}^* in the following way

$$\sum_{k=1}^{N+1} w_{mk}^* = DT^* - \sum_{n=1}^{N+1} \tilde{\alpha}_{mn}^\ell$$

for each row m of the matrix A .

We simply choose for each row $w_{m,N+1}^* = DT^* - \sum_{n=1}^{N+1} \tilde{\alpha}_{mn}^\ell$, and we use the leftmost rule to extract shape matrices with associated decomposition coefficients. It is a linear time algorithm and another proof of optimality of the sweep technique. Indeed, instead of $w_{m,N+1}^*$ we can choose any variable therefore, for any feasible decomposition time \hat{DT} upper bound on the end position mn is

$$\tilde{\alpha}_{mn}^\ell + \hat{DT} - \sum_{k=1}^{N+1} \tilde{\alpha}_{mk}^\ell, \tilde{\alpha}_{mn}^r + \hat{DT} - \sum_{k=1}^{N+1} \tilde{\alpha}_{mk}^r,$$

for the left and right end positions, respectively.

Finally, as a result of our work, we get the following algorithm.

Algorithm 3.6.1 (Unconstrained DT Problem)

Input: A

Output: *Decomposition of A*

(1.) *Compute DT^* using (3.26)*

(2.) *Compute matrices \tilde{L} , \tilde{R}*

(3.) *Extract shape matrices using the leftmost rule*

Chapter 4

Decomposition Cardinality

Problem

4.1 NP-hardness

While the decomposition time problem is solvable in linear time, the (unconstrained) decomposition cardinality problem $\min\{DC(\alpha) : A = \sum_{k \in \mathcal{K}} \alpha_k Y^k\}$ turns out to be *NP*-hard. This was proved by Burkard(2002) [14] for matrices with at least two rows using a reduction from subset sum. In the following we will strengthen his result.

Proposition 4 *If the given matrix consist of a single row then there exists a solution which is optimal to both problems DT and DC.*

Proof: Consider any optimal solution of DC problem. If there are two intervals such that the left end of one interval and the right end of the other one are at the same position, i.e. $[a, b)$ and $[b, c)$ with corresponding coefficients α and β , respectively. If $\alpha \geq \beta$, then we replace these intervals by the following intervals $[a, c)$ and $[a, b)$ with coefficients β and $\gamma = \alpha - \beta$, respectively. If it is not the case, i.e. $\alpha < \beta$, then we replace them by $[a, c)$ and $[b, c)$ with coefficients α and $\gamma = \beta - \alpha$, respectively. By merging the intervals we do not increase the number of shape matrices but we reduce the decomposition time by $\min\{\alpha, \beta\}$. We can do this until all intervals have different end positions, which can be done in linear time. Then by the invariant property of the decomposition $w_n = 0$, for all positions, i.e. the resulting decomposition is optimal to DT. \square

But it is not the case for matrices with more than 2 rows.

Example 4.1.1 Consider matrix from Example 3.4.1.

$$A = \begin{pmatrix} 5 & 10 & 6 \\ 4 & 1 & 1 \\ 7 & 0 & 0 \end{pmatrix}.$$

Now consider the following decomposition of the given matrix A .

$$\begin{aligned} Y^1 &= Y \begin{pmatrix} [1, 2) \\ [2, 4) \\ [1, 2) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_1 = 1 \\ Y^2 &= Y \begin{pmatrix} [1, 3) \\ [1, 2) \\ [1, 1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \alpha_2 = 4 \\ Y^3 &= Y \begin{pmatrix} [2, 4) \\ [2, 2) \\ [1, 2) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_3 = 6 \end{aligned}$$

with $DC = 3$ and $DT = 11$. From the Example 3.4.1 we know that the minimal decomposition time for this matrix is $DT^* = 10$. Therefore, for any optimal solution to DT problem of the given matrix A , $w_{1n}^* = 0$, for all $n = 1, \dots, 4$, since sum of elements in first row of matrix \tilde{L} is 10, i.e. for any matrices L and R corresponding to a decomposition of A , the first rows of these matrices the same as in matrices \tilde{L} and \tilde{R} , respectively.

$$\tilde{L} = \begin{pmatrix} 5 & 5 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{R} = \begin{pmatrix} 0 & 0 & 4 & 6 \\ 0 & 3 & 0 & 1 \\ 0 & 7 & 0 & 0 \end{pmatrix}$$

Then doing little observation on the first rows of matrices L and R , one can conclude that there is no decomposition of A with $DT = 10$ and $DC = 3$.

Theorem 8 [4] *The C1 decomposition cardinality problem is strongly NP-hard, even for matrices with a single row.*

Proof: The decision version of the C1 decomposition cardinality problem is as follows:

C1 Decomposition-Cardinality DC

Input: Matrix $A = (a_1, \dots, a_N)$, $K \in \mathbb{N}$

Output: Does there exist a decomposition of A into at most K C1 (row) matrices?

We reduce the following well-known strongly NP-complete problem [23] to DC.

Three Partitioning (3-PART)

Input: $B, Q \in \mathbb{N}$; $b_1, \dots, b_{3Q} \in \mathbb{N}$ with $\sum_{j=1}^{3Q} b_j = QB$ and $\frac{B}{4} < b_j < \frac{B}{2}$

Output: Does there exist a partitioning of $\{b_1, \dots, b_{3Q}\}$ into triples T_1, \dots, T_Q such that $\sum_{b \in T_q} b = B$ for all $q = 1, \dots, Q$?

We define

$$\begin{aligned} N &:= 4Q, \\ a_n &:= \begin{cases} \sum_{j=1}^n b_j, & \text{if } n \leq 3Q \\ (4Q - n + 1)B, & \text{if } n > 3Q, \end{cases} \\ K &:= 3Q. \end{aligned} \tag{4.1}$$

Claim: DC has YES output \iff 3-PART has YES output.

“ \Leftarrow ” For $j = 1, \dots, 3Q$ let $q \in \{1, \dots, Q\}$ be such that $b_j \in T_q$. A feasible output for DC is given by intervals $[j, 3Q + q + 1)$, $j = 1, \dots, 3Q$ and $\alpha_j = b_j$ (see Figure 4.1).

“ \Rightarrow ” By the definition of a_n , A cannot have a decomposition with cardinality smaller than $3Q$ since $b_j > 0$, $j = 1, \dots, 3Q$. Consider a solution of DC given by intervals $I_q = [l_q, r_q)$ and coefficients α_q , $q = 1, \dots, 3Q$, such that the sum of the interval lengths is maximized. We derive the following properties.

1. Due to Proposition 4 we may assume that for all $p, q \in \{1, \dots, 3Q\}$ $\ell_q \neq r_p$. Otherwise we can merge intervals as we did in the proof of Proposition 4.
2. Without loss of generality $\ell_q = q$ for all $q = 1, \dots, 3Q$. This follows since $a_1 < a_2 < \dots < a_{3Q}$ and some interval has to start in q .

3. $r_q > 3Q$ for all $q = 1, \dots, 3Q$. Otherwise, we have a contradiction to 1 and 2 with $l_p = r_q$ for some $p = 1, \dots, 3Q$.
4. $r_q \neq 3Q + 1$ for all $q = 1, \dots, 3Q$. Otherwise some $l_q = 3Q + 1$ would be needed since $a_{3Q} = a_{3Q+1}$. This would contradict 2.

Hence all intervals end in the set $\{3Q + 2, \dots, 4Q + 1\}$. Define triples T_1, \dots, T_Q by

$$b_j \in T_q \Leftrightarrow r_j = 3Q + j + 1.$$

By definition of a_{3Q+j} , the sum of the $b_j \in T_q$ equals B . This is obviously true for $j = Q$, since $a_{3Q+Q} = a_{4Q} = B$. For $j = Q - 1, \dots, 1$ this follows by an inductive argument. □

$(b_1, \dots, b_{3Q}) = (9, 11, 10, 11, 9, 12, 14, 9, 11)$	
$A = (9, 20, 30, 41, 50, 62, 76, 85, 96, 96, 64, 32)$	
$T_1 = \{9, 11, 12\}$	$Y^1 = Y([1, 11]) \quad \alpha_1 = 9$ $Y^2 = Y([2, 11]) \quad \alpha_1 = 11$ $Y^3 = Y([6, 11]) \quad \alpha_1 = 12$
$T_2 = \{10, 11, 11\}$	$Y^4 = Y([3, 12]) \quad \alpha_1 = 10$ $Y^5 = Y([4, 12]) \quad \alpha_1 = 11$ $Y^6 = Y([9, 12]) \quad \alpha_1 = 11$
$T_3 = \{9, 14, 9\}$	$Y^7 = Y([5, 13]) \quad \alpha_1 = 9$ $Y^8 = Y([7, 13]) \quad \alpha_1 = 14$ $Y^9 = Y([8, 13]) \quad \alpha_1 = 9$

Figure 4.1: 3-PART \propto DC with $B = 32, Q = 3, N = 12, K = 9$, and A are computed according to (4.1).

Corollary 5 *For any L and R matrices the problem of finding the optimal decomposition (with respect to the DC objective) is strongly NP-hard.*

Proof: Follows from Theorem 8 and Proposition 4. □

4.2 Polynomially solvable Cases

In some cases DC, however, can be solved in polynomial time. We need the following proposition to find some easily solvable instances of the decomposition cardinality problem.

Proposition 5 *If A is a positive integer multiple of an integer matrix B , i.e. $A = pB$, $p \geq 0$ and integer, then for the decomposition time problem the integer multiple of an optimal decomposition of B is also an optimal decomposition for the matrix A .*

Proof: Obviously, the integer multiple of any decomposition of B is a decomposition of A and $\tilde{A} = p\tilde{B}$. Therefore, for the unconstrained case, the statement follows immediately from 3.26. For the constrained case, observe that if we neglect the integrality of the coefficients α_k then the statement follows from the (DT-IP) formulation with respect to A and B . On the other hand, Algorithms 3.4.1 yield an integer solution only due to integrality of the input matrix. This completes the proof. \square

Proposition 6 *If A is a positive integer multiple of a binary matrix then the C1 decomposition cardinality problem can be solved in polynomial time for the constrained and unconstrained case.*

Proof: Observe that for binary matrices, $DT(\alpha) = DC(\alpha)$ since α_k is binary for all $k \in \mathcal{K}'$. Hence, if matrix A is a binary matrix then we can use Algorithm 3.4.1 (for the unconstrained case Algorithm 3.6.1) to solve the decomposition cardinality problem.

Let A be an integer multiple of a binary matrix B , i.e., $A = pB$. Then from any decomposition of B , multiplying by p , we get a decomposition of A with the same cardinality. Therefore, if B yields an optimal solution of the DC problem for A then using Algorithm 3.4.1 (for the unconstrained case Algorithm 3.6.1) we can find in polynomial time a decomposition of B and consequently a decomposition of A . We complete the proof by showing that for any decomposition of A there exists a decomposition of B with the same or a smaller cardinality. Consider any decomposition $A = \sum_{k=1}^K \alpha_k Y^k$. Observe that, if $\alpha_k = p$ for all $k = 1, \dots, K$ then we have a decomposition of B with the same cardinality. If not, we can assume without loss of generality that

$$A = \sum_{k=1}^{k_0} \alpha_k Y^k + p \sum_{k=k_0+1}^K Y^k$$

where $\alpha_k < p$ for all $k = 1, \dots, k_0$. Let A' and B' be matrices defined as

$$A' := A - p \sum_{k=k_0+1}^K Y^k = \sum_{k=1}^{k_0} \alpha_k Y^k$$

$$B' := B - \sum_{k=k_0+1}^K Y^k.$$

Then $A' = pB'$ since B is binary and $A = pB$. Consider any optimal DT decomposition of B'

$$B' := \sum_{k=1}^{k_1} \bar{Y}^k.$$

Note that for B' , $DT = DC = k_1$. Then from Proposition 5 follows that $pk_1 \leq \sum_{k=1}^{k_0} \alpha_k$, which implies that $k_1 < k_0$ since by our assumption $\alpha_k < p$ for all $k = 1, \dots, k_0$. Therefore, the decomposition of B

$$B = \sum_{k=k_0+1}^K Y^k + \sum_{k=1}^{k_1} \bar{Y}^k.$$

has smaller cardinality than the decomposition of A . □

4.3 IP model

Consider the general formulation of the problem.

$$\begin{aligned} \min \quad & \sum_{i \in \Omega} \beta_i \\ \text{s.t.} \quad & \sum_{i \in \Omega} \alpha_i S_i = A \end{aligned} \tag{4.2}$$

$$\begin{aligned} \alpha_i &\leq \mathbb{M} * \beta_i & \forall i \in \Omega \\ \alpha_i &\in \mathbb{Z}_+ & \forall i \in \Omega \\ \beta_i &\in \{0, 1\} & \forall i \in \Omega \end{aligned} \tag{4.3}$$

where \mathbb{M} is a sufficiently large number, for example the largest element of matrix A ; and Ω is the index set of all shape matrices.

In this section we develop IP model with smaller number of variables for a DC problem. We can always bound the optimal number of shape matrices from above, say by Q , using heuristic algorithms mentioned in Chapter 2 or in Chapter 6. It is obvious that $Q \leq M * N$, since all binary matrices with only one none zero element are shape matrices and any matrix A can be decomposed using those

matrices. Any shape matrix can be presented by using two binary matrices with exactly one non zero element in each row of those matrices, where each none zero element of one matrix presents the left end of intervals and the other presents the right end of intervals. On the other hand any pair of binary matrices, say $LP = (\ell_{mn})_{M \times N+1}$ and $RP = (r_{mn})_{M \times N+1}$, which satisfy the following conditions present a shape matrix.

$$\sum_{j=1}^{N+1} \ell_{mj} = 1 \quad \forall m \in \mathcal{M} \quad (4.4)$$

$$\sum_{j=1}^{N+1} r_{mj} = 1 \quad \forall m \in \mathcal{M} \quad (4.5)$$

$$\sum_{j=1}^k \ell_{mj} - \sum_{j=1}^k r_{mj} \geq 0 \quad \forall k \in \mathcal{N} \quad \forall m \in \mathcal{M} \quad (4.6)$$

$$\sum_{j=1}^k \ell_{mj} - \sum_{j=1}^k r_{m+1,j} \geq 0 \quad \forall k \in \mathcal{N} \quad \forall m \in \mathcal{M} \setminus M \quad (4.7)$$

$$\sum_{j=1}^k \ell_{m+1,j} - \sum_{j=1}^k r_{mj} \geq 0 \quad \forall k \in \mathcal{N} \quad \forall m \in \mathcal{M} \setminus M \quad (4.8)$$

where

- Conditions (4.4) and (4.5) require that each row of the matrices LP and RP must have exactly one none zero element.
- (4.6) presents the condition that left end of an interval must be to the left of right end of the interval
- Interleaf motion constraints are presented by conditions (4.7) and (4.8).

Thus we can formulate the DC problem in terms of Q pairs of binary matrices (LP_q, RP_q) , $q \in \mathcal{Q}$, which satisfy (4.4)–(4.8) To be short with notations we denote by \mathcal{Q} the index set $1, \dots, Q$. Recall that due to Theorem 1 we can represent the intensity cover constraint (4.2) in terms of total time used by the end positions in the decomposition, (2.2). Let us denote by α_q the coefficients associated with (LP_q, RP_q) , $q \in \mathcal{Q}$. Then, the following matrices $XL_q = (x\ell_{mn}^q)_{M \times N+1}$ and $XR_q = (xr_{mn}^q)_{M \times N+1}$ present the total time used by end positions corresponding

to the shape matrix presented by (LP_q, RP_q)

$$\sum_{j=1}^{N+1} x \ell_{mj}^q = \alpha_q \quad \forall m \in \mathcal{M} \quad (4.9)$$

$$\sum_{j=1}^{N+1} x r_{mj}^q = \alpha_q \quad \forall m \in \mathcal{M} \quad (4.10)$$

$$x \ell_{mn}^q \leq \mathbb{M} * \ell_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \quad (4.11)$$

$$x r_{mn}^q \leq \mathbb{M} * r_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \quad (4.12)$$

Then due to Theorem 1 and invariant property of decomposition, Q shape matrices represented by pairs of binary matrices (LP_q, RP_q) , $q \in \mathcal{Q}$ satisfies the intensity cover constraint (4.2) if and only if there exist $w_{mn} \geq 0$, $\forall m \in \mathcal{M}$, $n \in \mathcal{N}$ such that the following equations are satisfied

$$\sum_{q=1}^Q x \ell_{mn}^q - w_{mn} = \tilde{\alpha}_{mn}^{\ell} \quad \forall m \in \mathcal{M}, n \in \mathcal{N}$$

$$\sum_{q=1}^Q x r_{mn}^q - w_{mn} = \tilde{\alpha}_{mn}^r \quad \forall m \in \mathcal{M}, n \in \mathcal{N}$$

where $\tilde{\alpha}_{mn}^{\ell}$ and $\tilde{\alpha}_{mn}^r$ are defined according to G1 from the given matrix A .

Therefore we get the following IP model for DC Problem

$$\min \quad \sum_{q=1}^Q \beta_q$$

s.t.

$$\beta_q = \sum_{j=1}^{N+1} \ell_{mj}^q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (4.13)$$

$$\beta_q = \sum_{j=1}^{N+1} r_{mj}^q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (4.14)$$

$$\sum_{j=1}^k \ell_{mj}^q - \sum_{j=1}^k r_{mj}^q \geq 0 \quad \forall k \in \mathcal{N} \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (4.15)$$

$$\sum_{j=1}^k \ell_{mj}^q - \sum_{j=1}^k r_{m+1,j}^q \geq 0 \quad \forall k \in \mathcal{N} \forall m \in \mathcal{M} \setminus M, \forall q \in \mathcal{Q} \quad (4.16)$$

$$\sum_{j=1}^k \ell_{m+1,j}^q - \sum_{j=1}^k r_{mj}^q \geq 0 \quad \forall k \in \mathcal{N} \forall m \in \mathcal{M} \setminus M, \forall q \in \mathcal{Q} \quad (4.17)$$

$$\sum_{j=1}^{N+1} x \ell_{mj}^q = \alpha_q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (4.18)$$

$$\sum_{j=1}^{N+1} x r_{mj}^q = \alpha_q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (4.19)$$

$$\sum_{q=1}^Q x \ell_{mn}^q - w_{mn} = \tilde{\alpha}_{mn}^{\ell} \quad \forall m \in \mathcal{M}, n \in \mathcal{N} \quad (4.20)$$

$$\sum_{q=1}^Q x r_{mn}^q - w_{mn} = \tilde{\alpha}_{mn}^r \quad \forall m \in \mathcal{M}, n \in \mathcal{N} \quad (4.21)$$

$$x \ell_{mn}^q \leq \mathbb{M} * \ell_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q} \quad (4.22)$$

$$x r_{mn}^q \leq \mathbb{M} * r_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q} \quad (4.23)$$

$$\alpha_q, x \ell_{mn}^q, x r_{mn}^q \in \mathbb{Z}_+ \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q}$$

$$\beta_q, \ell_{mn}^q, r_{mn}^q \in \{0, 1\} \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q}$$

$$w_{mn} \geq 0 \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$$

4.4 Ordering, Fixing and Line search scheme

Linear and Lagrangian relaxation of the IP is very weak due to the big \mathbb{M} ; and consequently it increases the computation time drastically. Thus it is essential to have a good lower bound. As we have seen in the proof of Theorem 1 each non-zero element of matrices L and R needs a corresponding matrix used in a decomposition of A . Therefore, the number of non-zero elements in each row of matrices \tilde{L} and \tilde{R} is as a lower bound of the decomposition cardinality problem. Consequently, the maximum of these lower bounds is the best one obtainable in this way for given matrix A . i.e.

$$DC(\alpha) \geq \Phi := \min \{k_0 : k_0 \geq |\{\alpha_{mn}^\delta : \alpha_{mn}^\delta \neq 0, n \in \mathcal{N}\}| \forall m \in \mathcal{M}, \delta \in \{\ell, r\}\}.$$

Adding the following inequality

$$\sum_{q=1}^Q \beta_q \geq \Phi$$

into IP we will improve the relaxation. However, Φ is not strong enough. In *IP* model the number of variables and constraints greatly depends on the number of shape matrices Q . If the gap between Q and the minimal number of shape matrices are big then then the IP solver "carrying on its shoulder" the burden of all additional constraints and variables at all branch–decision nodes. To deal with those difficulties we use the line search method to solve the problem.

Algorithm 4.4.1 (Line Search Scheme)

Let Lb, Ub be the upper and lower bound on the minimal number of shape matrices

- (1.) *Consider line segment $[b, c]$ where $Lb \leq b \leq c \leq Ub$*
 - (2.) *Solve IP on the $[b, c]$*
 - (3.) **If** *IP is not feasible* **then** $Lb := c + 1$; *GO TO (1.)*
 - (4.) **else If** $DC^* < c$ **then** *Optimal solution is found GO TO (6.);*
 - (5.) $Lb := c$; *GO TO (1.)*
 - (6.) *Output*
-

In addition, in our IP formulation of the problem the shape matrices are *impartial*, i.e. the order of shape matrices are arbitrary. We may assume that the shape matrices are ordered by the decomposition coefficients. Which decreases the number of the nodes in branching scheme. But, before that we can fix some of the binary variables. As mentioned before, each non-zero element of matrices \tilde{L} and \tilde{R} needs a corresponding shape matrix used in a decomposition of A . Therefore, we choose the row with maximal number of non-zero elements, say k , among rows of matrices \tilde{L} and \tilde{R} ; and fix corresponding variables of the first k shape matrices. For example, assume that row m of \tilde{L} has the maximum number of non-zero elements $\tilde{\alpha}_{mi_1}^\ell, \tilde{\alpha}_{mi_2}^\ell, \dots, \tilde{\alpha}_{mi_k}^\ell$ then for first k shape matrices we can fix some of the values of variables in rows m as shown in the next:

$$\begin{aligned} \ell_{mn}^q &= \begin{cases} 1 & \text{if } n = i_q \\ 0 & \text{otherwise} \end{cases} & \forall n \in \mathcal{N} \\ r_{mn}^q &= 0 & \forall n < i_q \end{aligned}$$

for all $q = 1, \dots, k$. Then we order the remaining shape matrices by the decomposition coefficients, i.e. we add the following inequalities to the IP model

$$\beta_q \geq \beta_{q+1}$$

for all $q = k + 1, \dots, Q$.

4.5 Numerical results

The model was tested using CPLEX 9.0 embedded in C++ on PC, AMD Athlon(tm) 64 Processor 2800, 1.81 GHz, 512 MB RAM. Unfortunately, we could solve only small sized problems with very small upper bound on the number of shape matrices. We could solve the problem only for 4×4 and 5×5 matrices when the Q is not greater than 5 within 0.7–600 sec. For problems solved by the CPLEX, the gap between optimal solution and LP relaxation were about 16.67%–52.9%. The poor relaxation caused the drastic increase of the branching nodes and cuts as well, when the number of integer variables increases.

One way to tackle the difficulty is the usage of polyhedral analysis. In order to find facets of the integer polyhedron of the IP model we used Polyhedron Representation Transformation Algorithm. Unfortunately, PORTA could not find the polyhedral description for a small instance within 48 hours and got stuck in the memory.

Next what we can try is to relax the integrality of the variables α_q , $x\ell_{mn}^q$, xr_{mn}^q and w_{mn} . If it is "easily" solvable than the IP model then the solution of the relaxed problem might be helpful to find feasible solution close to the optimal solution.

The relaxation is mixed integer program (MIP) with binary variables. For this problem we will use new technique developed recently, accepted in journal of Discrete and Applied Mathematics in 2005, for MIP problems with binary variables and big M .

4.6 Bender's Combinatorial Cut

In 2005 Gianni Codato and Matteo Fischetti introduced new technique, based on Hooker's idea of deriving Bender's cuts from minimal sets of inconsistencies, proposed in ([31]), so called "***Bender's combinatorial Cut***" for MIP's involving logical implications modelled through big M coefficients. In this section we introduce the main idea of the method and for further details see [15]. The solution mechanism resembles the Bender's decomposition method but the cuts generated from slave problem are purely combinatorial and do not depend on the value of big- M used in the MIP formulation. Let be given a MIP problem with the following structure:

$$\begin{aligned}
 \min \quad & c^T x \\
 \text{s.t.} \quad & Fx \geq g \\
 & Dy \geq e \\
 & Mx_i + a_i y \geq b_i \quad \forall i = 1, \dots, n \\
 & x \in \{0, 1\}^n \\
 & y \in \mathbb{R}^m
 \end{aligned}$$

where $a_i \in \mathbb{R}^m$ and M is sufficiently large number.

The linking between the binary variables x and the continuous variables y is only due to constraints where the big- M is involved.

They split the MIP problem into two sub problems:

- MASTER

$$\begin{aligned}
 \min \quad & c^T x \\
 \text{s.t.} \quad & Fx \geq g \\
 & x \in \{0, 1\}^n
 \end{aligned}$$

- SLAVE, a linear system parametrized by x

$$\begin{aligned}
 Dy & \geq e \\
 a_i y & \geq b_i - Mx_i \quad \forall i = 1, \dots, n \\
 y & \in \mathbb{R}^m
 \end{aligned}$$

Let us solve the master problem. If this problem is infeasible then the original MIP problem is infeasible as well. Otherwise, let x^* be an optimal solution to master problem. We use x^* as a parameter for the slave problem. If the slave problem has a feasible solution, say y^* , then clearly (x^*, y^*) is an optimal solution to MIP problem. If it is not the case, i.e. slave problem is infeasible, then we cut the optimal solution x^* off from the master problem. For this propose Codato and Fischetti introduced so called Bender's combinatorial cuts in the following way. Consider any infeasible subsystem of the slave problem. Let C be the index set of variables x_i for which the corresponding constraint

$$a_i y \geq b_i - Mx_i^*$$

is included in the subsystem. Indeed, if C is an empty set then the original MIP problem is infeasible. Assume that, C is not empty then observe that at least one of the values x_i^* of the variables x has to be changed in order to make the subsystem feasible, that is the Combinatorial Bender's cut:

$$\sum_{i \in C: x_i^* = 0} x_i + \sum_{i \in C: x_i^* = 1} (1 - x_i) \geq 1 \quad (4.24)$$

One or more cuts of this type can be generated from the current optimal solution of the master problem and added to the master problem. And this procedure repeated again further exploring the polyhedral description of the IP problem.

4.7 Decomposition

Using the Bender's decomposition for the relaxation of the IP problem we get the following subproblems.

- MASTER Problem

$$\begin{aligned}
\min \quad & \sum_{q=1}^Q \beta_q \\
\text{s.t.} \quad & \beta_q = \sum_{j=1}^{N+1} \ell_{mj}^q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \\
& \beta_q = \sum_{j=1}^{N+1} r_{mj}^q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \\
& \sum_{j=1}^k \ell_{mj}^q - \sum_{j=1}^k r_{mj}^q \geq 0 \quad \forall k \in \mathcal{N} \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \\
& \sum_{j=1}^k \ell_{mj}^q - \sum_{j=1}^k r_{m+1,j}^q \geq 0 \quad \forall k \in \mathcal{N} \forall m \in \mathcal{M} \setminus M, \forall q \in \mathcal{Q} \\
& \sum_{j=1}^k \ell_{m+1,j}^q - \sum_{j=1}^k r_{mj}^q \geq 0 \quad \forall k \in \mathcal{N} \forall m \in \mathcal{M} \setminus M, \forall q \in \mathcal{Q} \\
& \beta_q, \ell_{mn}^q, r_{mn}^q \in \{0, 1\} \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q}
\end{aligned}$$

- SLAVE Problem, a linear system parametrized by an optimal solution of the master problem

$$\begin{aligned}
& \sum_{j=1}^{N+1} x \ell_{mj}^q = \alpha_q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \\
& \sum_{j=1}^{N+1} x r_{mj}^q = \alpha_q \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \\
& \sum_{q=1}^Q x \ell_{mn}^q - w_{mn} = \tilde{\alpha}_{mn}^{\ell} \quad \forall m \in \mathcal{M}, n \in \mathcal{N} \\
& \sum_{q=1}^Q x r_{mn}^q - w_{mn} = \tilde{\alpha}_{mn}^r \quad \forall m \in \mathcal{M}, n \in \mathcal{N} \\
& x \ell_{mn}^q \leq \mathbb{M} * \ell_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q} \quad (4.25) \\
& x r_{mn}^q \leq \mathbb{M} * r_{mn}^q \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q} \quad (4.26) \\
& \alpha_q, x \ell_{mn}^q, x r_{mn}^q \in \mathbb{Z}_+ \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall q \in \mathcal{Q} \\
& w_{mn} \geq 0 \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}
\end{aligned}$$

We replace the constraints (4.25) and (4.26) by the following constraints according to an optimal solution of the master problem in the following way:

$$\begin{aligned}
x\ell_{mn}^q &= 0 & \text{if } \ell_{mn}^q &= 0 \\
x\ell_{mn}^q &\geq 1 & \text{if } \ell_{mn}^q &= 1 \\
xr_{mn}^q &= 0 & \text{if } \ell_{mn}^q &= 0 \\
xr_{mn}^q &\geq 1 & \text{if } r_{mn}^q &= 1
\end{aligned}$$

The "complicated" constraints (4.25) and (4.26), where the big- \mathbb{M} is involved, were not only presenting the link between binary variables and continuous variables, they also present the relation between binary variables in terms of continuous variables. By using Benders decomposition we have cut the relationship between binary variables. Thus, the master problem further will be decomposed into Q independent problems.

$$\begin{aligned}
\min \quad & \beta_q \\
\text{s.t.} \quad & \\
& \beta_q = \sum_{j=1}^{N+1} \ell_{mj}^q \\
& \beta_q = \sum_{j=1}^{N+1} r_{mj}^q \\
& \sum_{j=1}^k \ell_{mj}^q - \sum_{j=1}^k r_{mj}^q \geq 0 \\
& \sum_{j=1}^k \ell_{mj}^q - \sum_{j=1}^k r_{m+1,j}^q \geq 0 \\
& \sum_{j=1}^k \ell_{m+1,j}^q - \sum_{j=1}^k r_{mj}^q \geq 0 \\
& \beta_q, \ell_{mn}^q, r_{mn}^q \in \{0, 1\}
\end{aligned}$$

Of course, using Bender's combinatorial cuts we can recover back the relation between binary variables. But from computational point of view this will be the most inefficient way to do it, since it will increase the number of constraints in the master problem dramatically.

4.8 Strengthen the Master Problem

It will be too ambitious if we want to recover almost "fully" the relation between binary variables. Of course, it would be nice, from computational point of view, if we can recover the relation "quite good" adding "small" number of constraints.

Observe that solving relaxation of the IP problem is equivalent to the following problem when the the given set Ω is the set of all possible shape matrices. From now on we will call this problem as *minimal cone problem*

MINIMAL CONE PROBLEM

Let be given a set of points $\Omega = \{S_1, S_2, \dots, S_k\}$ in \mathbb{R}_+^p and a point $A \in \mathbb{R}_+^p$. Find a cone with minimal dimension which is generated from a subset of Ω and contains the given point A .

Let us denote by C_Ω a cone C generated by a subset of Ω . Then making the relation of binary variables stronger in the master problem is equivalent to finding a polyhedron that contains all binary vectors whose components present a subset of Ω and the cone generated from this subset C_Ω has $\dim C_\Omega \leq Q$ and $A \in C_\Omega$.

Example 4.8.1 *If a shape matrix consists of 1 row and 2 columns; and $Q = 2$ then a point $(0, 1, 1, 0)$ presents a set of shape matrices $S_1 = (0, 1)$ and $S_2 = (1, 0)$ which correspond to the first and last two components of the given point, respectively.*

The first thing which we could try is to find the set which contains all cones C_B with $\dim C_B \leq Q$ and contains A . Where B is the set of all points in positive orthant with components not greater than 1. Unfortunately, this set is not a convex set at all(see the next example), i.e. it is not a polyhedron.

Example 4.8.2 *Let $Q=2$ and $A = (2, 1)$ then the following points $(0, 1, 1, 0)$ and $(1, 0, 1, 1)$ are in the set which we are looking for.*

$$S_1 = (0, 1), S_2 = (1, 0) \text{ and } S_1 + S_2 = (0, 1) + 2(1, 0) = (2, 1) = A$$

$$S_3 = (1, 0), S_4 = (1, 1) \text{ and } S_3 + S_4 = (1, 0) + (1, 1) = (2, 1) = A$$

Consider convex combination of these points with $\lambda = 1/2$, i.e.

$$\frac{1}{2}(0, 1, 1, 0) + \frac{1}{2}(1, 0, 1, 1) = \left(\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}\right)$$

Then the cone generated by points $(\frac{1}{2}, \frac{1}{2})$ and $(1, \frac{1}{2})$ does not contain the given point $A = (2, 1)$.

That means the condition of decomposability itself can not be written using linear system of equations and inequalities. Then the next thing is how to find a "small enough" polyhedron which contains all binary vectors whose components present a subset of Ω and the cone generated from this subset C_Ω has $\dim C_\Omega \leq Q$ and $A \in C_\Omega$; and "small" number of integral points which does present decomposition of the given point(matrix) A . Unfortunately, at the best of my knowledge, there is no work done related with minimal cone problem. We do not know how to find such a polyhedron therefore we will start from the polyhedron that contains all points whose associated cone contains the given point A . Then using Bender's combinatorial cuts we improve the description of this set. If a cone generated by a set of shape matrices S^1, S^2, \dots, S^Q contains the given matrix A then

$$a_{mn} > 0 \Rightarrow \sum_{q=1}^Q S_{mn}^q \geq 1 \quad (4.27)$$

We use the polyhedron defined by the set of inequalities (4.27), defined according non-zero elements of the matrix A ; to tighten the relation between shape matrices. We call this polyhedron as a "component cover" polyhedron.

As we did for IP formulation of the DC we fix variables with respect to non zero elements of the row with maximal number of nonzero elements among rows of matrices \tilde{L} and \tilde{R} .

Moreover, we order the remaining shape matrices by the order of left end positions. Assume that the row with maximal number of nonzero elements has p non-zero elements then we add the following constraints

$$\sum_{n=1}^k \ell_{1n}^q - \sum_{n=1}^k \ell_{1n}^{q+1} \leq 0 \quad \forall k \in \mathcal{N} \quad (4.28)$$

for all $q = p, \dots, Q - 1$.

4.9 Irreducible Inconsistent Subset

Bender's combinatorial cut separates the current optimal solution of the master problem along with other combinatorial possible solutions of the master problem which do not satisfy the Bender's cut. In order to improve efficiency of the cut, Codato and Fischetti([15]) suggested to use the minimal subset of indexes C to generate a Bender's cut.

Definition 4 *Irreducible Inconsistent Subsystem(IIS) of system of linear equations and inequalities is a subsystem which is infeasible and further elimination of arbitrary one of the equations or inequalities breaks the infeasibility of the subsystem.*

Finding the minimal IIS(MIIS) leads to the following minimal weighted IIS problem where each conditional constraints (4.25) and (4.26) weighted by 1 and the remaining constraints of the slave problem weighted by 0. Minimal weighted IIS problem turned out to be \mathcal{NP} -hard problem [20, 24]. Therefore, we use the following heuristics to find the MIIS:

- Find IIS of the slave problem
- Remove all conditional constraints, which are in IIS, from the slave problem
- Do the above procedure until the remaining slave problem is feasible
- Choose the minimal C

It is worth to mention that without constraints (4.25) and (4.26), the slave problem is feasible for any instance of the given matrix A . From computational point of view it was efficient to add along with Bender's cut generated by the smallest IIS several other Bender's cuts in to the master problem, at the same time.

4.10 Numerical result

We used the line search to improve the computational efficiency of the method as we did for IP formulation of the DC problem. After each solution process on a line segment we removed all Bender's cuts, which are generated on this line segment, from the master problem. All tests are done using CPLEX 9.0 embedded in C++ on PC, AMD Athlon(tm) 64 Processor 2800, 1.81 GHz, 512 MB RAM. We could solve only small sized problems with small upper bound on the number of shape matrices, $M, N \leq 5$, with $Q \leq 7$. Computation time for solved instances was between 3.16–760.3 sec. In all tests the number of the Bender's combinatorial cuts increased very fast and steeply; and the master problem became soon very large. Consequently, the overloading the master problem slowed down the node processing significantly. Unfortunately, we do not have good mechanism to purge the master problem by removing from time to time some of the cuts from the master problem. In the future, the computation time of our approach can be improved in the following ways

- to find better polyhedron than component cover polyhedron used in our model

- to find good mechanism to prevent the master problem from overload
- to improve "quality" of the Bender's cuts using better heuristics for MIIS
- to use heuristics to generate Bender's cuts from LP relaxation of the master problem

Chapter 5

Heuristics

As we have seen in the previous chapter, the DC problem is strongly \mathcal{NP} -hard and different approaches could only find the optimal solution of the problem only for small sized instances in practically acceptable computation time. Therefore, it is essential to have a heuristic algorithm which can find a "good" solution to the DC problem within practically acceptable computation time.

5.1 DT Solution Based Greedy Heuristic

5.1.1 Constrained DC

We propose the following "greedy" algorithm based on the intuitive idea that "if decomposition time $DT(\alpha)$ is small and coefficients of the decomposition are on average high then decomposition cardinality is small". Based on this, first we solve (DT-IP) to find matrices L and R , which yield the minimum $DT(\alpha)$, then each time we extract a shape matrix with maximum possible coefficient such that the residual of matrices L and R again present a decomposition. Recall that in the proof of Theorem 1 and consequently in the Algorithm 3.4.1 we used leftmost non-zero elements in the rows of L and R , which preserve conditions (2.3), (2.4) and (2.7). If for any extraction of a shape matrix these conditions are maintained then the residual matrices represent a decomposition.

Let us introduce $(M - 1) \times N$ matrices \bar{A} and \hat{A} defined as follows

$$\bar{a}_{mn} = \sum_{k=1}^n \alpha_{mk}^{\ell} - \sum_{k=1}^n \alpha_{m+1,k}^r \quad (5.1)$$

$$\hat{a}_{mn} = \sum_{k=1}^n \alpha_{m+1,k}^{\ell} - \sum_{k=1}^n \alpha_{mk}^r \quad (5.2)$$

for all $m \in \mathcal{M} \setminus \{M\}$ and $n \in \mathcal{N} \setminus \{N + 1\}$.

Then conditions (2.3), (2.4) and (2.7) for residual matrices of L , R and A can be written in terms of \bar{A} and \hat{A} , respectively, as

$$\bar{a}_{mn} \geq \alpha \quad \forall n : \ell_m \leq n < r_{m+1}, \quad \forall m \in \mathcal{M} \setminus \{M\} \quad (5.3)$$

$$\hat{a}_{mn} \geq \alpha \quad \forall n : \ell_{m+1} \leq n < r_m, \quad \forall m \in \mathcal{M} \setminus \{M\} \quad (5.4)$$

$$a_{mn} \geq \alpha \quad \forall n : \ell_m \leq n < r_m, \quad \forall m \in \mathcal{M}, \quad (5.5)$$

where α is the coefficient corresponding to the extracted shape matrix $Y([\ell_m, r_m])_{m \in \mathcal{M}}$. Therefore, in order to extract a shape matrix in a greedy way, we need to solve the following problem ($\max -\alpha$).

$$\begin{aligned} & \max \quad \alpha \\ & \text{s.t. (5.3), (5.4), (5.5)} \end{aligned}$$

$$\ell_m \leq r_{m+1} \quad \forall m \in \mathcal{M} \setminus \{M\} \quad (5.6)$$

$$\ell_{m+1} \leq r_m \quad \forall m \in \mathcal{M} \setminus \{M\} \quad (5.7)$$

$$\ell_m \leq r_m \quad \forall m \in \mathcal{M} \quad (5.8)$$

$$\alpha_{m\ell_m}^{\ell} \geq \alpha \quad \forall m \in \mathcal{M} \quad (5.9)$$

$$\alpha_{mr_m}^r \geq \alpha \quad \forall m \in \mathcal{M} \quad (5.10)$$

$$\ell_m, r_m \in \mathcal{N} \quad \forall m \in \mathcal{M}.$$

Algorithm 5.1.1 (Greedy Approach to the Constrained DC Problem)

Input: Matrix A

Output: Decomposition of A into shape matrices

(1.) Compute $DT(\alpha)$ and matrices L and R using Algorithm 3.4.2

(2.) Compute \bar{A} , \hat{A} according to (5.1) and (5.2)

(3.) Set $k := 0$ and initialize
 $\alpha := \min_{m \in \mathcal{M}, \delta \in \{\ell, r\}} \max_{n \in \mathcal{N}} \alpha_{mn}^\delta$

(4.) While $DT(\alpha) \neq 0$ do

(4.1.) If $\alpha \neq 1$
then For $m = 1$ to M do
 $I_m := \{[p, q] : (5.5), (5.8) - (5.10)\}$
If $I_m = \{\emptyset\}$ then GO TO (4.9.)
end For

(4.2.) $m := 1$

(4.3.) While $m \neq M$ do
If $m \leq 1$ then
 $m := 1$
 $[\ell_1, r_1] := \text{lexmin}\{[p, q] : [p, q] \in I_1\}$
end If
Remove intervals $[p, q]$ with $q < \ell_m$ from I_{m+1}
If $I_{m+1} = \{\emptyset\}$ then GO TO (4.9.)
Find $\text{lexmin}\{[p, q] : [p, q] \in I_{m+1}\}$ such that

- $\ell_m \leq q, \quad p \leq r_m$
- $\bar{a}_{mn} \geq \alpha$ for all $n : \ell_m \leq n < q$
- $\hat{a}_{mn} \geq \alpha$ for all $n : p \leq n < r_m$

If such an interval $[p, q]$ exists
then $m := m + 1$
 $[\ell_{m+1}, r_{m+1}] := [p, q]$
else
 $I_m := I_m \setminus \{[\ell_m, r_m]\}$
If $I_m = \{\emptyset\}$ then GO TO (4.9.)
else $m := m - 1$

end If

end while (4.3.)

(4.4.) *Set $k := k + 1$*

(4.5.) *Extract shape matrix $Y^k = Y([\ell_m, r_m])_{m \in \mathcal{M}}$*
 with coefficient $\alpha_k := \alpha$

(4.6.) *Update $A, L, R, \bar{A}, \hat{A}$*

(4.7.) *Set $DT(\alpha) := DT(\alpha) - \alpha_k$*

(4.8.) *For $m = 1$ to M do*
 Remove intervals which do not satisfy
 (5.5), (5.9) and (5.10) from I_m
 end for
 If $I_m \neq \{\emptyset\}$ for all $m \in \mathcal{M}$ then GO TO (4.2.)
 else extract shape matrices until $DT(\alpha) = 0$
 (use leftmost nonzero elements of L and R)
 end If

(4.9.) *$\alpha := \alpha - 1$*

(4.10.) *end while (4.)*

End.

The algorithm 5.1.1 considers iteratively all possible values of α in the while loop (4.). Whenever $\alpha = 1$, i.e., the maximal possible coefficient is 1 for any extraction the number of shape matrices is equal to the decomposition time. Therefore, the algorithm uses the leftmost non-zero elements of matrices L and R to extract shape matrices. If $\alpha \neq 1$ then in each iteration for each row m it constructs the set of intervals I_m defined by conditions (5.5), (5.8)–(5.10). In the while loop (4.3.) these sets are iteratively reduced with respect to conditions (5.3), (5.4), (5.6) and (5.7) such that the first elements of these sets form a shape matrix or some of the sets become empty.

If there exists a shape matrix the algorithm extracts it and repeats the procedure again to find the next shape matrix with the same coefficient. When there is no possibility to extract a shape matrix with coefficient α , α is updated in (4.9.) and the above procedure is repeated for the new value of α .

Example 5.1.1 Consider the matrix A given in Example 3.4.1. Using the matrices

$$L = \begin{pmatrix} 5 & 5 & 0 & 0 \\ 4 & 3 & 0 & 3 \\ 7 & 0 & 0 & 3 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 0 & 4 & 6 \\ 0 & 6 & 0 & 4 \\ 0 & 7 & 0 & 3 \end{pmatrix}$$

found in this example, we can compute

$$\bar{A} = \begin{pmatrix} 5 & 4 & 4 \\ 4 & 0 & 0 \end{pmatrix}, \quad \hat{A} = \begin{pmatrix} 4 & 7 & 3 \\ 7 & 1 & 1 \end{pmatrix}.$$

According to (3.), (4.1.)–(4.2.) the minimum of the maximal elements in rows of matrices L and R is $\alpha = 4$ and we get

$$I_1 = \{[1, 3], [1, 4], [2, 3], [2, 4]\}$$

$$I_2 = \{[1, 2]\}$$

$$I_3 = \{[1, 2]\}.$$

The first elements of these sets satisfy conditions (5.3), (5.4), (5.6) and (5.7), i.e., we can extract a shape matrix

$$Y^1 = Y \begin{pmatrix} [1, 3] \\ [1, 2] \\ [1, 2] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_1 = 4.$$

Updating interval sets, using conditions (5.5), (5.9) and (5.10), with respect to residual matrices

$$A = \begin{pmatrix} 1 & 6 & 6 \\ 0 & 1 & 1 \\ 3 & 0 & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 5 & 0 & 0 \\ 0 & 3 & 0 & 3 \\ 3 & 0 & 0 & 3 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 0 & 0 & 6 \\ 0 & 2 & 0 & 4 \\ 0 & 3 & 0 & 3 \end{pmatrix}$$

we get $I_2 = I_3 = \{\emptyset\}$. Considering the next possible value of $\alpha = 3$ we get the

following sets

$$\begin{aligned} I_1 &= \{[2, 4]\} \\ I_2 &= \{[4, 4]\} \\ I_3 &= \{[1, 2], [4, 4]\}. \end{aligned}$$

We can exclude $[1, 2]$ from I_3 , since it does not satisfy conditions (5.6) and (5.7) with $[4, 4] \in I_2$. The remaining intervals form the shape matrix

$$Y^2 = Y \begin{pmatrix} [2, 4] \\ [4, 4] \\ [4, 4] \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \alpha_2 = 3$$

which satisfy (5.3) and (5.4). Repeating the above procedure we get

$$Y^3 = Y \begin{pmatrix} [2, 4] \\ [2, 2] \\ [1, 2] \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_3 = 2$$

$$Y^4 = Y \begin{pmatrix} [1, 4] \\ [2, 4] \\ [1, 2] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \alpha_4 = 1.$$

So we have an alternative decomposition of A with a smaller number of shape matrices compared with the decomposition in Example 3.4.1.

5.1.2 Unconstrained case

We can use our greedy approach for unconstrained DC problems since we can compute the matrices L and R as follows:

$$L = \tilde{L} + W \quad R = \tilde{R} + W$$

where W is any matrix with positive integer entries such that

$$\sum_{n \in \mathcal{N}} w_{mn} = DT - DT_m$$

for all $m \in \mathcal{M}$. Note that the same decomposition time DT results from every row of the matrix. For instance, we can choose W such that each row m of W has no more than one non-zero element which corresponds to the maximum element among the elements of the corresponding rows of \tilde{L} and \tilde{R} and has a value equal to $DT - DT_m$. Therefore, no optimization problem is solved to find W and we only need to find α by solving problem $(\max -\alpha)$, which now becomes

$$\begin{aligned} & \max \quad \alpha \\ \text{s.t.} \quad & (5.5), (5.8), (5.9), (5.10) \\ & \ell_m, r_m \in \mathcal{N} \quad \forall m \in \mathcal{M}. \end{aligned}$$

Thus, the greedy algorithm for the the unconstrained decomposition cardinality problem is as follows.

Algorithm 5.1.2 (Greedy Approach to Unconstrained DC Problem)

Input: Matrix A

Output: Decomposition of A into $C1$ matrices

(1.) Compute $DT, DT_m, m \in \mathcal{M}$ and matrices L and R

(2.) Initialize $\alpha := \min_{m \in \mathcal{M}, \delta \in \{\ell, r\}} \max_{n \in \mathcal{N}} \alpha_{mn}^\delta$

(3.) Set $k := 0$

(4.) While $DT \neq 0$ do

For $m = 1$ to M do

$I_m := \{[p, q] : (5.5), (5.8) - (5.10)\}$

If $I_m = \{\emptyset\}$ then GO TO (4.7.)

end For

(4.1.) If $\alpha \neq 1$

then

(4.2.) Set $k := k + 1$

(4.3.) Extract $C1$ matrix $Y^k = Y([\ell_m, r_m])_{m \in \mathcal{M}}$

with coefficient $\alpha_k := \alpha$

where $[\ell_m, r_m]$ is the first element of I_m

(4.4.) *Update A, L, R*

(4.5.) *Set $DT := DT - \alpha_k$*

(4.6.) *For $m = 1$ to M do*
 Remove intervals which do not satisfy
 (5.5), (5.9) and (5.10) from I_m
 end for
 If $I_m \neq \{\emptyset\} \forall m \in \mathcal{M}$ then GO TO (4.2.)
 else extract C1 matrices until $DT = 0$
 (use first elements of $I_m, m \in \mathcal{M}$)

(4.7.) $\alpha := \alpha - 1$

(4.8.) *end while (4.)*

End.

5.2 Numerical results

The algorithm tests are done using CPLEX 9.0 embedded in C++ on PC, AMD Athlon(tm) 64 Processor 2800, 1.81 GHz, 512 MB RAM. In the next Table 5.1 the numerical results are shown in comparison with other existing algorithm designed for DC problem (Siochi's algorithm and improved Xia-Verhey's algorithm as it was used by Boland, Hamacher and Lenzen in [8]). For the test we have used over 1000 randomly generated data with intensity level below 16.

	Number of Shape Matrices			Total Beam-on Time		
	Alg.5.1.2	Siochi	XV-BHL	Alg. 5.1.2	Siochi	XV-BHL
5x5	6.34	9.3	8.98	23.41	26.7	33.54
10x10	18.1	27.5	17.81	41.8	52.4	85.61
20x20	30.6	53.9	35.68	77.99	90.63	133.69
30x30	50.9	72.04	53.16	113.9	155.42	199.29
40x40	70.6	121.51	71.03	140.5	199.34	266.11

Figure 5.1: Comparison of the algorithms

Solution found by Algorithm 5.1.1 was in all instances better than Siochi's and in almost 90% better than XV-BHL algorithm.

The computation time is shown in the next table

	CPU Time sec.
5x5	0.0
10x10	0.1
20x20	1.05
30x30	2.5
40x40	4.4

Figure 5.2: Average computation time for different matrix size

5.3 Flexible Minimal DT Based Heuristic

We have seen that DT solution based heuristics, introduced in previous section, yield "quite" good solution to DC problem in comparison with other existing algorithms. However, there is still a possibility to improve the solutions obtained by those algorithms. In those algorithms we have used the matrices L and R obtained by solving the DT problem, i.e. we have fixed the possible end positions of the intervals. Therefore, we do not really extract the shape with the maximal possible decomposition coefficient which can be used in the decomposition of the given matrix A within minimal decomposition time DT. Let us denote the optimal value of the DT problem by DT^* . Here, we extract the shape matrices in pure greedy way, i.e. without fixing the end positions, within the minimal decomposition time.

Let us consider the following MIP problem for some fixed positive integer value of C :

$$\begin{aligned}
 \max \quad & \min DT(D) \\
 \text{s.t.} \quad & C * D + B = A \\
 & \min DT(D) + \min DT(B) = DT^* \\
 & d_{mn} \geq 0 \text{ integer } \forall i = 1, \dots, N; j = 1, \dots, N \\
 & b_{mn} \geq 0 \forall i = 1, \dots, N; j = 1, \dots, N
 \end{aligned} \tag{5.11}$$

where

- $D = \{d_{mn}\}_{M \times N}$, $B = \{b_{mn}\}_{M \times N}$
- $\min DT(D)$ and $\min DT(B)$ are minimum decomposition times corresponding to the matrices D and B , respectively.

Proposition 7 *Suppose that C is the maximum possible coefficient which can be used in a decomposition of the intensity matrix A . Let matrices D^* and B^* be defined by an optimal solution of (5.11). Then in any integer decompositions of the matrices D^* and B^**

$$\begin{aligned}
 \sum_{i=1}^p \alpha_i * S_i = D^* & \quad \text{s.t.} \quad \sum_{i=1}^p \alpha_i = \min DT(D^*), \\
 \sum_{j=1}^q \beta_j * S_j = B^* & \quad \text{s.t.} \quad \sum_{j=1}^q \beta_j = \min DT(B^*)
 \end{aligned}$$

the coefficients satisfy the following

1. $\alpha_i = 1 \quad \forall i = 1, \dots, p$
2. $\beta_j < C \quad \forall j = 1, \dots, q.$

Proof: Follows immediately from the assumptions. □

Then, $\min DT(D^*)$ presents the maximum number of shape matrices, with decomposition coefficient C , which can be used in a decomposition of A within the decomposition time DT^* . Moreover, D^* is a sum of such matrices used in a decomposition and $\min DT(D^*) = \min DC(D^*)$. Recall that, the last condition holds for binary matrices, however, D^* is not necessary to be a binary matrix. Therefore, we can use methods for DT problem to solve the DC problem for the matrix D^* .

Our greedy approach is equivalent to solving a sequence of problems (5.11) for each values of C starting from the maximum element of A down to 2.

Due to the Theorem 1, we can represent matrices D and B by a pair of matrices (L^D, R^D) and (L^B, R^B) , respectively.

In terms of matrices $L^D = \{\ell_{mn}^D\}$, $R^D = \{r_{mn}^D\}$, $L^B = \{\ell_{mn}^B\}$ and $R^B = \{r_{mn}^B\}$, due to the Theorem 1, the problem (5.11) can be formulated in the following way:

$$\begin{aligned}
\max \quad & \sum_{n=1}^{N+1} \ell_{1n}^D \\
\text{s.t.} \quad & \\
C * \sum_{j=1}^n (\ell_{mj}^D - r_{mj}^D) + \sum_{j=1}^n (\ell_{mj}^B - r_{mj}^B) = a_{mn} & \quad (5.12)
\end{aligned}$$

$$\forall m \in \mathcal{M}, n \in \mathcal{N} \setminus \{N+1\}$$

$$\sum_{j=1}^n (\ell_{mj}^D - r_{mj}^D) \geq 0 \quad \forall m \in \mathcal{M}, n \in \mathcal{N} \setminus \{N+1\} \quad (5.13)$$

$$\sum_{j=1}^{N+1} (\ell_{mj}^D - r_{mj}^D) = 0 \quad \forall m \in \mathcal{M} \quad (5.14)$$

$$\sum_{j=1}^n (\ell_{mj}^D - r_{m+1,j}^D) \geq 0 \quad \forall m \in \mathcal{M} \setminus M, n \in \mathcal{N} \quad (5.15)$$

$$\sum_{j=1}^n (\ell_{m+1,j}^D - r_{m,j}^D) \geq 0 \quad \forall m \in \mathcal{M} \setminus M, n \in \mathcal{N} \quad (5.16)$$

$$\sum_{j=1}^{N+1} \ell_{mj}^D = \sum_{n=1}^{N+1} \ell_{1n}^D \quad \forall m \in \mathcal{M} \setminus \{1\} \quad (5.17)$$

$$\sum_{j=1}^n (\ell_{mj}^B - r_{mj}^B) \geq 0 \quad \forall m \in \mathcal{M}, n \in \mathcal{N} \setminus \{N+1\} \quad (5.18)$$

$$\sum_{j=1}^{N+1} (\ell_{mj}^B - r_{mj}^B) = 0 \quad \forall m \in \mathcal{M} \quad (5.19)$$

$$\sum_{j=1}^n (\ell_{mj}^B - r_{m+1,j}^B) \geq 0 \quad \forall m \in \mathcal{M} \setminus M, n \in \mathcal{N} \quad (5.20)$$

$$\sum_{j=1}^n (\ell_{m+1,j}^B - r_{m,j}^B) \geq 0 \quad \forall m \in \mathcal{M} \setminus M, n \in \mathcal{N} \quad (5.21)$$

$$\sum_{j=1}^{N+1} \ell_{mj}^B = \sum_{n=1}^{N+1} \ell_{1n}^B \quad \forall m \in \mathcal{M} \setminus \{1\} \quad (5.22)$$

$$\sum_{n=1}^{N+1} \ell_{1n}^D + \sum_{n=1}^{N+1} \ell_{1n}^B = DT^* \quad (5.23)$$

$$\ell_{mn}^D, r_{mn}^D \geq 0 \quad \text{integer} \quad \forall m \in \mathcal{M}, n \in \mathcal{N}$$

$$\ell_{mn}^B, r_{mn}^B \geq 0 \quad \forall m \in \mathcal{M}, n \in \mathcal{N}$$

Where,

(5.12) presents the intensity cover constraints, due to (2.6)

(5.13)–(5.18) present the leaf collision constraints,

(5.21)–(5.23) present decomposition time of the matrices D, B and A, respectively.

Using Algorithm 3.5.2 we can find upper bounds for the leaves in any row m and position n. Thus, for some of the variables we can fix the values in the following way:

$$\text{if } \bar{\ell}_{mn} < C \quad \text{then } \ell_{mn}^D = 0 \quad (5.24)$$

$$\text{if } \bar{r}_{mn} < C \quad \text{then } r_{mn}^D = 0 \quad (5.25)$$

where $\bar{\ell}_{mn}$ and \bar{r}_{mn} are upper bounds at position mn for the left leaf and the right leaf, respectively. We propose the following algorithm for DC Problem

Algorithm 5.3.1 (Flexible DT Solution Based Algorithm)

Input: Matrix A

Output: Decomposition of A

(1.) Solve DT Problem; Find DT^* , Let $MaxElement = \max a_{mn}$.

(2.) for ($C=MaxElement$; $C > 1$; $C --$)

{ Use Algorithm 3.5.2 and for some variables fix the values
Solve problem (5.11).

Extract shape matrices from L^D and R^D using leftmost rule

$DT^* = DT^* - C * DT(D)$;

$A = A - C * D$;

$C = C - 1$; }

if ($B \neq 0$) Extract shape matrices from B using leftmost rule

End.

5.4 Numerical Result

The algorithm tests are done using CPLEX 9.0 embedded in C++ on PC, AMD Athlon(tm) 64 Processor 2800, 1.81 GHz, 512 MB RAM. In the next table the numerical results are shown in comparison with XV-BHL algorithm which was competitive with our greedy Algorithm 5.1.1. Tests are done on randomly generated 15x15 matrices with different intensity levels up to 16. For each row of the Table 5.3 the results were averaged on over 500 tests. Then it shows the average beam-on time (BOT), average number of shape matrices (NS), average CPU time (CPU) and their maximal and minimal values.

L	Algorithm 5.3.1									XV-BHL	
	BOT	NS	CPU	\overline{BOT}	\underline{BOT}	\overline{NS}	\underline{NS}	\overline{CPU}	\underline{CPU}	BOT	NS
3	11.3	10.6	0.8	13	9	12	9	2.13	0.4	20.6	13.7
4	15.5	12.1	1.8	20	14	14	11	3.31	1.04	19.8	13.3
5	19.2	13.7	2.9	23	16	16	12	12.89	1.48	47.5	20.3
6	23.5	15.1	3.5	28	20	18	13	10.74	1.93	47.4	20.2
7	28.3	16.8	3.8	34	21	19	15	6.37	2.64	47.2	20.3
8	31.4	17.5	4.6	38	27	20	15	6.17	3.06	46.3	19.9
9	36.7	18.9	4.8	46	32	21	16	8.04	3.41	97.6	26.7
10	38.9	19.5	22.7	50	33	22	17	512.88	4.11	100.9	26.8
11	45.7	20.2	6.8	57	38	22	19	13.41	4.7	102.1	27.0
12	46.1	20.7	6.6	51	40	24	19	9.68	4.91	101.4	26.9
13	51.8	21.5	7.8	61	45	24	20	10.12	5.93	101.8	26.9
14	56.3	22.3	8.5	66	45	26	19	17.2	6.61	102.2	27.1
15	58.6	22.7	8.7	69	51	26	19	11.49	6.56	100.5	26.7
16	64.4	23.5	11.5	72	56	26	21	41.42	7.47	99.0	26.5

Figure 5.3: Comparison of the algorithms

Solution found by Algorithm 5.3.1 was significantly improved in comparison to the solutions found by Algorithm 5.1.1.

For an unconstrained case deleting the constraints related with the leaf collision, we get the following MIP problem:

$$\begin{aligned}
& \max \quad \sum_{n=1}^{N+1} \ell_{1n}^D \\
& \text{s.t.} \\
& C * \sum_{j=1}^n (\ell_{mj}^D - r_{mj}^D) + \sum_{j=1}^n (\ell_{mj}^B - r_{mj}^B) = a_{mn} & \forall m \in \mathcal{M}, n \in \mathcal{N} \setminus \{N+1\} \\
& \sum_{j=1}^n (\ell_{mj}^D - r_{mj}^D) \geq 0 & \forall m \in \mathcal{M}, n \in \mathcal{N} \setminus \{N+1\} \\
& \sum_{j=1}^{N+1} (\ell_{mj}^D - r_{mj}^D) = 0 & \forall m \in \mathcal{M} \\
& \sum_{j=1}^{N+1} \ell_{mj}^D = \sum_{n=1}^{N+1} \ell_{1n}^D & \forall m \in \mathcal{M} \setminus \{1\} \\
& \sum_{j=1}^n (\ell_{mj}^B - r_{mj}^B) \geq 0 & \forall m \in \mathcal{M}, n \in \mathcal{N} \setminus \{N+1\} \\
& \sum_{j=1}^{N+1} (\ell_{mj}^B - r_{mj}^B) = 0 & \forall m \in \mathcal{M} \\
& \sum_{j=1}^{N+1} \ell_{mj}^B = \sum_{n=1}^{N+1} \ell_{1n}^B & \forall m \in \mathcal{M} \setminus \{1\} \\
& \sum_{n=1}^{N+1} \ell_{1n}^D + \sum_{n=1}^{N+1} \ell_{1n}^B = DT^* \\
& \ell_{mn}^D, r_{mn}^D \geq 0 \quad \text{integer} & \forall m \in \mathcal{M}, n \in \mathcal{N} \\
& \ell_{mn}^B, r_{mn}^B \geq 0 & \forall m \in \mathcal{M}, n \in \mathcal{N}
\end{aligned}$$

For some variables we can fix the values as we did in 5.24 and 5.25 using Algorithm 3.5.2. The computation time is shown in the next table 5.4. Results shown in each row were averaged on over 10000 randomly generated 15x15 matrices with intensity level 16. The results are compared with XV-BHL.

L	Algorithm 5.3.1 for Unconstrained DC									XV-BHL	
	BOT	NS	CPU	\overline{BOT}	\underline{BOT}	\overline{NS}	\underline{NS}	\overline{CPU}	\underline{CPU}	BOT	NS
3	9.93	8.32	0.1156	13	8	10	7	0.29	0.09	16.26	10.93
4	14.13	9.95	0.2135	18	12	11	9	0.26	0.18	17.3	11.61
5	17.88	11.27	0.3068	23	15	13	10	0.38	0.28	34.31	15.62
6	21.48	12.1	0.4092	26	18	14	11	0.47	0.36	37.72	16.63
7	25.65	13.06	0.5066	34	21	16	12	0.6	0.45	40.22	17.55
8	29.17	13.73	0.6092	36	25	16	12	0.7	0.54	40.5	17.66
9	33.22	14.6	0.7103	42	28	16	13	0.79	0.65	67.14	21.05
10	36.69	15.01	0.8071	47	30	17	13	1.02	0.73	74.6	21.69
11	41.01	15.64	0.9056	51	34	18	14	1.01	0.81	79.39	22.47
12	44.29	16.01	1.0115	57	37	18	14	1.13	0.94	82.03	22.67
13	49.2	16.5	1.1082	61	39	18	15	1.23	0.99	86.07	23.6
14	52.79	16.94	1.2154	64	46	19	15	1.32	1.08	85.75	23.45
15	55.87	17.23	1.309	73	45	19	16	1.45	1.21	86.82	23.73
16	60.74	17.77	1.4225	81	50	20	16	1.58	1.3	87.43	23.9

Figure 5.4: Comparison of the algorithms

Chapter 6

Conclusion

In this thesis we have discussed the problem of decomposing an integer matrix A into a weighted sum $A = \sum_{k \in \mathcal{K}} \alpha_k Y^k$ of 0-1 matrices with the strict consecutive ones property. We have developed algorithms to find decompositions which minimize the decomposition time $\sum_{k \in \mathcal{K}} \alpha_k$ and the decomposition cardinality $|\{k \in \mathcal{K} : \alpha_k > 0\}|$.

In the absence of additional constraints on the 0-1 matrices Y^k we have given an algorithm that finds the minimal decomposition time in $\mathcal{O}(NM)$ time. For the case that the matrices Y^k are restricted to shape matrices – a restriction which is important in the application of our results in radiotherapy – we have given an $\mathcal{O}(NM^2)$ algorithm. This is achieved by solving an integer programming formulation of the problem by a very efficient combinatorial algorithm.

In addition, we have shown that the problem of minimizing decomposition cardinality is strongly NP-hard, even for matrices with one row (and thus for the unconstrained as well as the shape matrix decomposition). Our greedy heuristics are based on the results for the decomposition time problem and produce better results than previously published algorithms.

Bibliography

- [1] R.K. Ahuja and H.W. Hamacher. Linear time network flow algorithm to minimize beam-on-time for unconstrained multileaf collimator problems in cancer radiation therapy. *Networks*, 45(1):36–41, 2004.
- [2] M. Alber and F. Nüsslin. IMRT optimization under constraints for static and dynamic MLC delivery. *Phys. Med. Biol.*, 46(12):3229–3239, 2001.
- [3] D. Baatar and H.W. Hamacher. New LP model for multileaf collimators in radiation therapy planning. In *Proceedings of the Operations Research Peripatetic Postgraduate Programme Conference ORP³, Lambrecht, Germany*, pages 11–29, 2003.
- [4] D. Baatar, H.W. Hamacher, M. Ehrgott, and G.J. Woeginger. Decomposition of integer matrices and multileaf collimator sequencing. *Discrete Applied Mathematics*. Accepted for publication, 2005.
- [5] J. Balog, L. Angelos, et al. Dose computation for a one dimensional MLC. <http://www-madrad.radiology.wisc.edu/tomo/benchtop/balog.html>.
- [6] N. Becker and J. Wahrendorf. *Atlas of cancer mortality in the Federal Republic of Germany 1981 - 1990*. Springer, Berlin, third edition, 1998.
- [7] J. A. Belli, R. G. Lane, S. M. Morrill, and I. I. Rosen. Treatment plan optimization using linear programming. *Med. Phys.*, 18(2):141–152, 1991.
- [8] N. Boland, H.W. Hamacher, and F. Lenzen. Minimizing beam-on time in cancer radiation treatment using multileaf collimators. *Networks*, 43(4):226–240, 2003.
- [9] K. Booth and G. Lucker. Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms. *Journal of Computer and System Science*, 13:335–379, 1976.
- [10] T.R. Bortfeld, A.L. Boyer, D.L. Kahler, and T.J. Waldron. X-ray field compensation with multileaf collimators. *International Journal of Radiation Oncology, Biology, Physics*, 28(3):723–730, 1994.

- [11] A. L. Boyer, Y. Chen, and C.-M. Ma. Calculation of x-ray transmission through a multileaf collimator. *Med. Phys.*, 27(8):1717–1726, 2000.
- [12] A. Brahme, P. Kaellman, and R. Svensson. An analytical solution for the dynamic control of multileaf collimators. *Phys. Med. Biol.*, 39(1):37–61, 1994.
- [13] R. Bruni. On exact selection of minimally unsatisfiable subformulae. *Annals of Mathematics and Artificial Intelligence*, 43(1):35–50, 2005.
- [14] R.E. Burkard. Open Problem Session, Oberwolfach Conference on Combinatorial Optimization, November 24-29, 2002, 2002.
- [15] G. Codato and M. Fischetti. Bender’s combinatorial cut. In *Integer Programming and Combinatorial Optimization: 10th International IPCO Conference, New York, NY, USA, June 7-11, 2004. Proceedings*, pages 118–195, 2004.
- [16] D. J. Convery and M. E. Rosenbloom. The generation of intensity-modulated fields of conformal radiotherapy by dynamic collimation. *Phys. Med. Biol.*, 37(6):1359–1374, 1994.
- [17] B. Davey, N. Boland, and P. J. Stuckey. Efficient intelligent backtracking using linear programming. *INFORMS Journal of Computing*, 14(4):373–386, 2002.
- [18] L. Drever and P. Dickof. Virtual misro mlc commissioning. *Journal of Applied Clinical Medical Physics*, 6(2), 2005.
- [19] M. N. Du, L. J. Verhey, D. Yan, C. X. Yu, and S. Zhou. Optimization of leaf positions when shaping a radiation field with a multileaf collimator. *Phys. Med. Biol.*, 40:305–308, 1995.
- [20] M. E. Pfetsch E. Amaldi and L. E. Trotter. On the maximum feasible subsystem problem, iiss and iis-hypergraphs. *Mathematical Programming*, 95(3):533–554, 2003.
- [21] R. J. Gallagher and E. K. Lee. Mixed integer programming optimization models for brachytherapy treatment planning. *American Medical Informatics Association Annual Fall Symposium*, pages 278–282, 1997.
- [22] J. M. Galvin, X. Chen, and R. M. Smith. Combining multileaf fields to modulate fluence distributions. *Int. J. Radiat. Oncol. Biol. Phys.*, 27:697–705, 1993.

- [23] M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., San Francisco, CA, 1979.
- [24] J. Gleeson and J. Ryan. Identifying minimally infeasible subsystems of in-. equalities. *ORSA Journal on Computing*, 2(1):61–63, 1990.
- [25] H.W. Hamacher and K.-H. Küfer. Inverse radiation therapy planing – A multiple objective optimization approach. *Discrete Applied Mathematics*, 118:145–161, 2002.
- [26] B. J. M. Heijmen and J. P. C. Santvort. Dynamic multileaf collimation without 'tongue-and-groove' underdosage effects. *Phys. Med. Biol.*, 41:2091–2105, 1996.
- [27] R. W. Hill, M. P. Carol, B. H. Curran, and J. P. Strait. Delivery of intensity modulated radiation therapy using Computer-Controlled MLCs with the CORVUS[®] Inverse treatment planning system. NOMOS Corporation, Sewickley, PA, 15143; <http://www.nomos.com/iccr97/rob.htm>.
- [28] J. Hoeffner. *New Methods for solving the inverse problem in radiation therapy planning*. PhD thesis, University of Kaiserslautern, Germany, 1996.
- [29] A. Holder. Designing radiotherapy plans with elastic constraints and interior points methods. Department of Mathematics, Trinity University, USA; <http://www.trinity.edu/aholder/research/oncology/papers.html>, 2000.
- [30] T. W. Holmes, A. R. Bleier, et al. The effect of MLC leakage on the calculation and delivery of intensity modulated radiation therapy. NOMOS Corporation, Sewickley, PA, 15143; <http://www.nomos.com/iccr97/tim.htm>.
- [31] J. N. Hooker and G. Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.
- [32] T. Kalinowski. An algorithm for optimal MLC field segmentation with interleaf collision constraints. In *Proceedings of the Operations Research Peripatetic Postgraduate Programme Conference ORP³, Lambrecht, Germany, 2003*.
- [33] S. Kamath, S. Sahni, J. Li, J. Palta, and S. Ranka. Leaf sequencing algorithms for segmented multileaf collimator. *Physics in Medicine and Biology*, 48(3):307–324, 2003.
- [34] R. G. Lane, S. M. Morrill, I. I. Rosen, and J. A. Wong. Dose volume considerations with linear programming optimization. *Med. Phys.*, 18(6):1201–1210, 1991.

- [35] M. Langer, V. Thai, and L. Papiez. Improved leaf sequencing reduces segments of monitor units needed to deliver IMRT using MLC. *Medical Physics*, 28:2450–58, 2001.
- [36] E. K. Lee. Planning implants of radionuclides for the treatment of prostate cancer: An application of mixed integer programming. *OPTIMA, Mathematical Programming Society Newsletter*, 61, 1999.
- [37] E. K. Lee, R. J. Gallagher, D. Silvern, and C. Wu. Treatment planning for brachytherapy: an integer programming model, two computational approaches and experiments with permanent prostate implant planning. *Phys. Med. Biol.*, 44:145–165, 1999.
- [38] E. K. Lee, Gallagher R. J., and M. Zaider. Planning implants of radionuclides for the treatment of prostate cancer: an application of mixed integer programming. *OPTIMA, Mathematical Programming Society Newsletter*, 61, 1999.
- [39] F. Lenzen. An integer programming approach to the multileaf collimator problem. Master’s thesis, University of Kaiserslautern, Germany, 2000.
- [40] G. Loi, E. Pignoli, et al. Design and characterization of a dynamic multileaf collimator. *Phys. Med. Biol.*, 43:3149–3155, 1998.
- [41] L. Luo, X. Wu, and Y. Zhu. Linear programming based on neural networks for radiotherapy treatment planning. *Phys. Med. Biol.*, 45:719–728, 2000.
- [42] T. Mackie, G. Olivera, P. J. Reckwerdt, and D. Shepard. IMRT optimization. <http://www.madr.d.radiology.wisc.edu/tomo/opimization>.
- [43] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Technical Report 94 -10, Mathematical Programming Technical Report, 1994.
- [44] R. Mohan. Field shaping for three-dimensional conformal radiation therapy and multileaf collimation. *Seminars in Radiation Oncology*, 5(2):86–99, 1995.
- [45] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1988.
- [46] S. Papatheodorou, J. C. Rosenwald, et al. Utilisation d’un collimateur multilames pour la production de faisceaux modulés en intensité”. *Cancer Radiothérapie*, 2:392–403, 1998.
- [47] K. Preiser, Th. Bortfeld, K. Hartwig, et al. Inverse strahlentherapieplanung für intensitätsmodulierte photonenfelder. *Der Radiologe*, 38:228–234, 1998.

- [48] S. Quach. Pruning radiosurgery plans. Trinity University, USA; <http://www.trinity.edu/aholder/research/oncology/papers.html>, 2000.
- [49] W. Que. Comparison of algorithms for multileaf collimator field segmentation. *Med. Phys.*, 26:2390–2396, 1999.
- [50] P. A. Rubin. Solving mixed integer classification problems by decomposition. *Annals of Operations Research*, 74:51–64, 1997.
- [51] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1987.
- [52] D. M. Shepard, M. C. Ferris, G. H. Olivera, and T. R. Mackie. Optimizing the delivery of radiation therapy to cancer patients. *SIAM Review*, 41(4):721–744, 1999.
- [53] R.A Siochi. Minimizing static intensity modulation delivery time using an intensity solid paradigm. *Int. J. Radiat. Oncol. Biol. Phys.*, 43:671–680, 1999.
- [54] J. Stoer and C. Witzgall. *Convexity and Optimization in Finite Dimensions I*. Springer Verlag, 1970.
- [55] R. Svensson, P. Källman, and A. Brahme. An analytical solution for the dynamic control of multileaf collimators. *Phys. Med. Biol.*, 39:37–61, 1994.
- [56] V. Tsurkov. *Large Scale Optimization—Problems and Methods*, volume 51 of *Applied Optimization*. Kluwer Academic Publishers, 2001.
- [57] J. P. C. van Santvoort and B. J. M. Heijmen. Dynamic multileaf collimation without ‘tongue-and-groove’ underdosage effects. *Phys. Med. Biol.*, 41:2091–2105, 1996.
- [58] A. Wambersie and R. A. Gahbauer. Medical application of electron linacs. XVIII International Linac Conference Geneva, Switzerland, August 26-30, 1996.
- [59] S. Webb. *The Physics of Three-Dimensional Radiation Therapy*. IOP, Amsterdam, 1993.
- [60] S. Webb. Configuration options for intensity-modulated radiation therapy using multiple static fields shaped by a multileaf collimator. *Phys. Med. Biol.*, 43:241–261, 1998.
- [61] 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. *Phys. Med. Biol.*, 43:1481–1495, 1998.

- [62] S. Webb. A new concept of multileaf collimator (the shuttling MLC)- an interpreter for high-efficiency IMRT. *Phys. Med. Biol.*, 45:3343–3358, 2000.
- [63] L. A. Wolsey. *Integer Programming*. John Wiley and Sons, 1998.
- [64] X. Wu and Y. Zhu. An optimization method for importance factors and beam weights based on genetic algorithms for radiotherapy treatment planning. *Phys. Med. Biol.*, 46:1085–1099, 2001.
- [65] P. Xia and L. Verhey. Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple segments. *Medical Physics*, 25:1424–1434, 1999.
- [66] C. X. Yu, D. Yan, M. N. Du, Z. Zhou, and L. J. Verhey. Optimizaion of leaf-positions when shaping a radiation field with a multileaf collimator. *Phys. Med. Biol.*, 40:305–308, 1995.
- [67] T. C. Zhu, B. E. Bjärngard, and H. Shackford. X-ray source and the output factor. *Med. Phys.*, 22(6):793–798, 1995.

Curriculum Vitae

1. Personal Data:

- Name: Davaatseren Baatar
- Date of Birth: 10 November 1969
- Place of Birth: Ulaanbaatar, Mongolia

2. Education

- August 25, 2005 Disputation
- 11/2002–09/2005 Ph.D study in Optimization, at the Department of Mathematics, University of Kaiserslautern, Germany. (funded by Graduiertenkolleg "Mathematik und Praxis")
- 10/2000–09/2002 M.Sc in Optimization, University of Kaiserslautern, Germany
- 09/1992–06/1994 B.Sc in Mathematics, National University of Mongolia
- 09/1988–06/1991 Bachelor's study at State University of Irkutsk, Russia

3. Professional Career

- 07/1994–10/2000 Lecturer, Department of Mathematics, School of Economic Studies, NUM