# Decomposition of Integer Matrices and Multileaf Collimator Sequencing

Davaatseren Baatar, Horst W. Hamacher[1]
Fachbereich Mathematik
Technische Universität Kaiserslautern
Postfach 3029
67653 Kaiserslautern, Germany
email {baatar,hamacher}@mathematik.uni-kl.de

Matthias Ehrgott
Department of Engineering Science
The University of Auckland
Private Bag 92019
Auckland, New Zealand
email m.ehrgott@auckland.ac.nz

Gerhard J. Woeginger
Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
email gwoegi@win.tue.nl

19th October 2004

[1]This research is partially supported by the New Zealand Ministry for Research, Science and Technology through a Julius von Haast fellowship and the Deutsche Forschungsgemeinschaft DFG, Grant No. HA 1795/7.

## Abstract

In this paper we consider the problem of decomposing an integer matrix into a weighted sum of binary matrices that have the strict consecutive ones property. This problem is motivated by an application in cancer radiotherapy planning, namely the sequencing of multileaf collimators to realize a given intensity matrix. In addition we also mention another application in the design of public transportation. We are interested in two versions of the problem, minimizing the sum of the coefficients in the decomposition (decomposition time) and minimizing the number of matrices used in the decomposition (decomposition cardinality). We present polynomial time algorithms for unconstrained and constrained versions of the decomposition time problem and prove that the (unconstrained) decomposition time problem is strongly $NP$-hard. For the decomposition cardinality problem, some polynomially solvable special cases are considered and heuristics are proposed for the general case.



**Keywords:** Decomposition of integer matrices, consecutive ones property, multileaf collimator sequencing, radiotherapy.

# 1 Introduction

**Definition 1.1.** 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.

Let $\mathcal{K}$ be an index set of all $M \times N$ consecutive ones matrices and $\mathcal{K}' \subset \mathcal{K}$. 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 $\alpha_k, k \in \mathcal{K}'$ and $M \times N$ C1 matrices $Y^k, k \in \mathcal{K}'$ such that

$$A = \sum_{k \in \mathcal{K}'} \alpha_k Y^k. \tag{1}$$

In the following, we often use $\mathcal{M} := \{1, \ldots, M\}$, $\mathcal{N} = \{1, \ldots, N+1\}$ For each of the C1 matrices $Y^k$ there exist $\ell^k_m \in \mathcal{N}, r^k_m \in \mathcal{N}$ such that $Y^k = (y^k_{mn})$ is given by

$$y^k_{mn} = 1 \Longleftrightarrow \ell^k_m \leq n < r^k_m \quad \forall m \in \mathcal{M}. \tag{2}$$

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

$$Y^k = Y([\ell^k_m, r^k_m))_{m \in \mathcal{M}}.$$

**Example 1.2.** 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

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

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., Booth and Lucker (1976)). For this reason our definition contains the word strict which we will, however, delete subsequently.

C1 decompositions can be used in various applications, two of which are introduced next.

**Application 1.3 (Radiation Therapy Planning).** In intensity modulated radiation therapy (IMRT) planning, $A$ is a matrix that describes the intensity distribution across a radiation beam. These **intensity matrices** can be found, for instance, with the multicriteria approach to radiation therapy planning of Hamacher and Küfer (2002). In Figure 1 some intensity matrices are shown as greyscale coded grids. Black represents no radiation, the lighter the color the higher the radiation intensity.

Radiation according to an intensity matrix is delivered by multileaf collimators (MLC). Radiation is blocked out by pairs of metal leaves moved into the beam from left and right (black areas in the three rightmost squares in Figure 2). It can, however, pass through the opening between

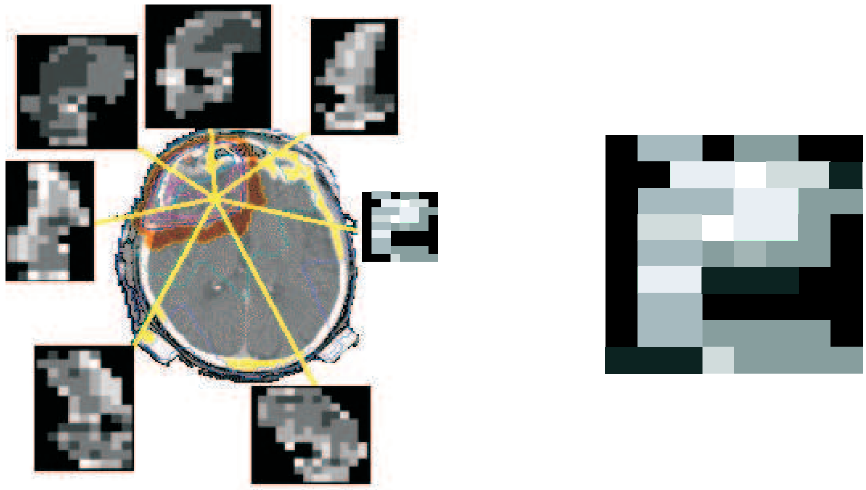
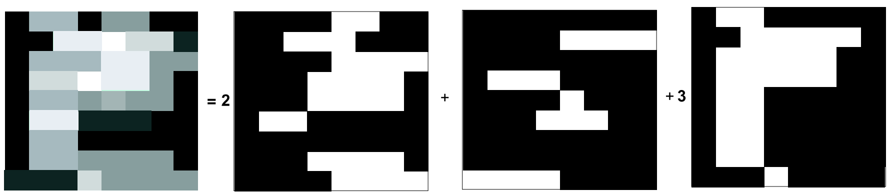


Figure 1: IMRT with intensity matrices represented as checker-board schemes.




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

Figure 2: Realization of an intensity matrix by overlaying radiation fields with different MLC configurations.

the leaves (white areas). By irradiating each of the MLC configurations for a certain amount of time (= intensity) the intensity matrix is realized.

Obviously, possible left/right leaf configurations can be represented by C1 matrices $Y^k$. If $\alpha_k$ is the duration of irradiation with a particular leaf configuration then (1) defines the realization of the intensity matrix. More details can be found, for instance, in Baatar and Hamacher (2003), Boland *et al.* (2003), Ahuja and Hamacher (2004), Engel (2003), Kalinowski (2003).

**Application 1.4 (Stop design in public transportation Hamacher *et al.* (2001), Schöbel *et al.* (2002), Ruf and Schöbel (2003)).** Consider a set $\mathcal{P}$ of customers and a set $\mathcal{S}$ of potential sites for installing a stop in a public transportation system. Assume it is required that each customer has a stop not further away than a distance $r$. This can be written as $dist(p,s) \le r$ for all $p \in \mathcal{P}$ with respect to some $s \in \mathcal{S}$.

The stop design problem can be written as a set covering problem as follows.

$$\begin{array}{rcl} \min\ cx & & \\ \text{s.t. } Ax & \geq & 1 \\ x & \in & \{0,1\}^{|\mathcal{S}|}, \end{array}$$

where $A = (a_{ps})_{\substack{p\in\mathcal{P}\\ s\in\mathcal{S}}}$ with $a_{ps} = \begin{cases} 1 & \text{if } dist(p,s) \leq r \\ 0 & \text{otherwise} \end{cases}$.

Obviously, the stops should be located along existing lines of the public transportation system. Figure 3 shows that, depending on the topology, $A$ may be C1 or not. The circles indicate all points at distance $r$ from customer $p$. Possible stop locations are indicated by crosses and are defined by intersections of the circles with the lines.

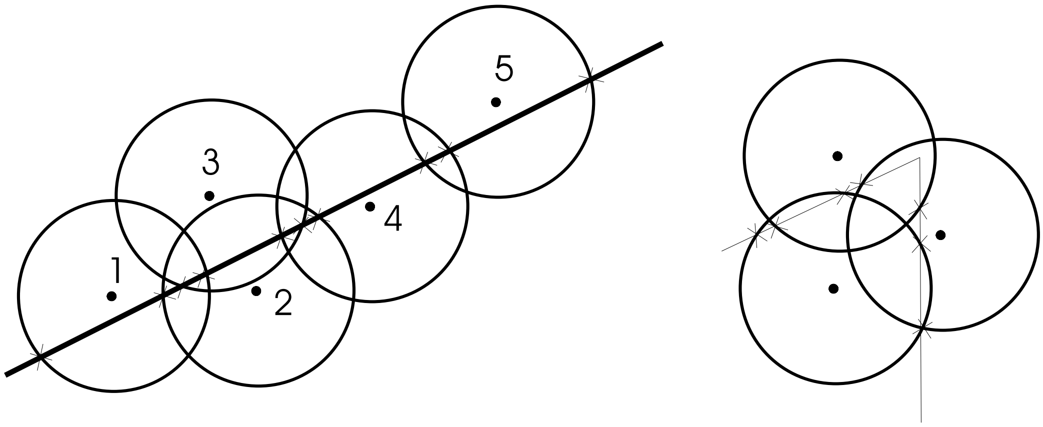



Figure 3: Two instances of the stop design problem.

The coefficient matrix of the first instance (left part of Figure 3),

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

is obviously C1. Since C1 matrices are totally unimodular (Nemhauser and Wolsey, 1988), the set covering problem is polynomially solvable. The coefficient matrix

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

of the second instance (right part of Figure 3) on the other hand is not C1 (neither in the strict sense used in this paper nor in the weak sense). It can, however, be written as a sum of C1 matrices – a fact which is used in Ruf and Schöbel (2003) to solve large instances of the set covering problem efficiently.

In this paper we consider two objective functions which can be used to evaluate a given C1 decompositions of type (1), the **decomposition time**

$$DT(\alpha) := \sum_{k \in \mathcal{K}'} \alpha_k \tag{3}$$

and the **decomposition cardinality**

$$DC(\alpha) := |\{\alpha_k : \alpha_k > 0\}|. \tag{4}$$

In the case where $\mathcal{K}' = \mathcal{K}$ (**unconstrained decomposition**) we will show in the next section that the minimization of $DT(\alpha)$ can be achieved in linear time. Specific choices of $\mathcal{K}'$ with important applications and resulting polynomial algorithms to minimize $DT(\alpha)$ are discussed in Section 3. That minimizing $DC(\alpha)$ defines an $NP$-hard problem is shown in Section 4. In that section we also present some ideas on heuristics for minimizing $DC(\alpha)$.

# 2 Linear Algorithm for Unconstrained Decomposition Time

In this section we assume throughout that $\mathcal{K}' = \mathcal{K}$, i.e., all C1 matrices are allowed in the decomposition (1). Note that the number of C1 matrices is exponential in the number of rows of $A$.

Given the integer 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

$$\tilde{a}_{mn} := a_{mn} - a_{m,n-1}. \tag{5}$$

Here $a_{m0} = a_{m,n+1} := 0$ for all $m \in \mathcal{M}$.

**Definition 2.1.** For $m \in \mathcal{M}$ let $\mathcal{P}_m := \{\ell : \ \tilde{a}_{m\ell} > 0\}$ and $\mathcal{Q}_m := \{r : \ \tilde{a}_{mr} < 0\}$. Then $\mathcal{L}_m := \{[\ell, r) : \ell \in \mathcal{P}_m,\ r \in \mathcal{Q}_m\}$ is the list of **crucial intervals** in the $C1$ decomposition.

**Lemma 2.2.** *For every $m \in M$*

$$DT_m := \sum_{l \in \mathcal{P}_m} \tilde{a}_{m\ell} = \sum_{r \in \mathcal{Q}_m} (-\tilde{a}_{mr}) \tag{6}$$

*is a lower bound for the decomposition time of the $m$-th row $A_m$ of $A$.*

*Proof.* Whenever $\tilde{a}_{m\ell} > 0$ any C1 decomposition needs to use intervals with left boundary in $\ell$ at least $\tilde{a}_{m\ell}$ times. Adding over $\mathcal{P}_m$ yields the result. It is clear that $\sum_{n \in \mathcal{N}} \tilde{a}_{mn} = 0$ for all $m$. □

Since any C1 decomposition of $A$ implies a C1 decomposition of its rows, we get the following result.

**Lemma 2.3.**

$$DT := \max_{m \in \mathcal{M}} DT_m \tag{7}$$

*is a lower bound for the decomposition time of $A$.*

Subsequently, we assume that every $\mathcal{L}_m$ is kept as a lexicographically sorted list and that $\mathcal{P}_m$ and $\mathcal{Q}_m$ are sorted in increasing order.

Next, we will show how crucial intervals can be extracted from $\mathcal{L}_m$ to get a minimum decomposition time algorithm for each row $A_m$ of $A$.

**Algorithm 2.4 (Extraction Procedure for Row Matrices).**

| | |
|---|---|
| **Input**: | *Row* $A_m$ *of* $A$ |
| | *Lists* $\mathcal{P}_m$ *and* $\mathcal{Q}_m$ |
| **Output**: | *Decomposition* $A_m = \sum_{k=1}^{K_m} \alpha_m^k Y_m^k$ *with minimal decomposition time,* |
| | $L_m$ *list of crucial intervals contributing to the decomposition* |
| *(1.)* | *Initialize* $k := 1$, $L_m := \emptyset$ |
| *(2.)* | *Choose first entry* $\ell$ *in* $\mathcal{P}_m$ *and first entry* $r \in \mathcal{Q}_m$ |
| | $L_m := L_m \cup \{[\ell, r)\}$ |
| *(3.)* | *Set* $Y_m^k = [\ell, r)$, $\alpha_m^k := \min\{\tilde{a}_{m\ell}, -\tilde{a}_{mr}\}$ |
| | $A_m = A_m - \alpha_m^k Y_m^k$ |
| *(4.)* | *If* $A_m = 0$ *output* $K_m := k$ |
| | *Remove* $\ell$ *from* $\mathcal{P}_m$ *if* $\alpha_m^k = \tilde{a}_{m\ell}$, *and* $r$ *from* $\mathcal{Q}_m$, *if* $\alpha_m^k = -\tilde{a}_{mr}$. |
| | *Set* $k := k+1$ *and goto (2.)* |

In each iteration at least one element is removed from $\mathcal{P}_m \cup \mathcal{Q}_m \subset \mathcal{N}$, such that the algorithm performs $\mathcal{O}(N)$ iterations. Since each iteration is done in constant time, the extraction procedure is a linear time algorithm. The resulting decomposition time is

$$DT(\alpha) = \sum_{k=1}^{K_m} \alpha_k = \sum_{\ell \in \mathcal{P}_m} \tilde{a}_{m\ell} = \sum_{r \in \mathcal{Q}_m} -\tilde{a}_{mr}$$

so that the lower bound of Lemma 2.2 is attained.

For each $m \in \mathcal{M}$ the output of Algorithm 2.4 includes a list $L_m$ of intervals which define the C1 row matrices used in the decomposition of row $A_m$. The next algorithm puts these intervals together to define a minimum decomposition time C1 decomposition of $A$. Since the decomposition times $DT(\alpha)$ of rows are in general different this putting together requires the usage of degenerate intervals $I_m = [\ell_m^k, r_m^k)$ with $\ell_m^k = r_m^k$.

**Algorithm 2.5 (Unconstrained Minimum C1 Decomposition Time).**

| | |
|---|---|
| **Input**: | *Integer Matrix* $A$ |
| **Output**: | *Decomposition* $A = \sum_{k=1}^{K_A} \alpha_k Y^k$ *with minimal decomposition time* |
| *(1.)* | *For* $m = 1, \ldots, M$ |
| |     *apply Algorithm 2.4 to obtain list* $L_m$ *of crucial intervals* |
| |     *with* $\alpha_m(I)$ $\forall I \in L_m$ |
| *(2.)* | *Set* $k = 0$ |
| *(3.)* | *While* $A \neq 0$ *do* |
| |   $k := k+1$ |
| |   *a) Choose* $I_m \in L_m$ $\forall m \in \mathcal{M}$ |
| |       *(where* $I_m = \emptyset$ *and* $\alpha(I_m) = \infty$ *if* $L_m = \emptyset$*)* |
| |   *b) Set* $Y^k := Y(I_1, ..., I_M)$ |
| |       $\alpha_k := \min_{m \in \mathcal{M}} \alpha(I_m)$ |
| |       $A := A - \alpha_k Y^k$ |
| |   *c) Set* $\alpha(I_m) := \alpha(I_m) - \alpha_k$ |
| |       $L_m := L_m \setminus \{I_m\}$ *if* $\alpha(I_m) = 0$ |
| *(4.)* | *Output* $K_A := k$ *and* $A = \sum_{k=1}^{K_A} \alpha_k Y^k$ |

If $m^* \in \mathcal{M}$ is an index in which the lower bound $DT$ of Lemma 2.3 is attained, $L_{m^*} \neq \emptyset$ will be maintained throughout the algorithm and thus $\sum_{k=1}^{K_A} \alpha_k = DT_{m^*} = DT$. Hence the algorithm provides an optimal solution of the minimum decomposition time problem. Its complexity is $\mathcal{O}(NM)$. It should be noted that a more efficient way to implement Algorithm 2.2 would include an update of DT starting $DT = \max_{m\in\mathcal{M}} DT_m$ until $DT = 0$ thus avoiding the time consuming update of $A$. An analogous observation holds for Algorithm 2.4.

**Example 2.6.** Consider $A = \begin{pmatrix} 3 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 3 & 5 \end{pmatrix}$. Thus

$$\tilde{A} = \begin{pmatrix} 3 & -1 & -2 & 0 & 1 & -1 \\ 1 & -1 & 0 & 3 & 2 & -5 \end{pmatrix}$$

with the lower bounds $DT_1 = 4$, $DT_2 = DT = 6$ from Lemmas 2.2 and 2.3. As output from Algorithm 2.4 we obtain

$$L_1 := \{[1,2),[1,3),[5,6)\} \text{ with } \alpha_1^1 = 1,\ \alpha_1^2 = 2,\ \alpha_1^3 = 1$$

and

$$L_2 := \{[1,2),[4,6),[5,6)\} \text{ with } \alpha_2^1 = 1,\ \alpha_2^2 = 3,\ \alpha_2^3 = 2.$$

Algorithm 2.5 provides the C1 decomposition

$$\begin{array}{rclcll}
Y^1 & = & Y\begin{pmatrix} [1,2) \\ [1,2) \end{pmatrix} & = & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, & \alpha_1 = 1 \\
Y^2 & = & Y\begin{pmatrix} [1,3) \\ [4,6) \end{pmatrix} & = & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, & \alpha_2 = 2 \\
Y^3 & = & Y\begin{pmatrix} [5,6) \\ [4,6) \end{pmatrix} & = & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, & \alpha_3 = 1 \\
Y^4 & = & Y\begin{pmatrix} \emptyset \\ [5,6) \end{pmatrix} & = & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, & \alpha_4 = 2
\end{array}$$

with $K_A = 4$ and $\sum_{k=1}^{K_A} \alpha_k = 6 = DT$. Note that any degenerate interval $[\ell_m^k, r_m^k)$ with $\ell_m^k = r_m^k$ can be used to present the empty set $\emptyset$.

It should be noted that the minimal C1 decomposition time would also be obtained by the network flow algorithm of Ahuja and Hamacher (2004). The latter algorithm and Algorithm 2.5 justify the "Sweep Algorithm" by Bortfeld *et al.* (1994) which is widely used in the sequencing of multileaf collimators for the realization of intensity matrices in radiation therapy (see Application 1.3) and which – to the best of our knowledge – was not proved to be optimal before 2003.

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

$$\begin{array}{rcl} \tilde{\alpha}_{mn}^{\ell} & = & \max\{0, a_{mn} - a_{m,n-1}\}, \\ \tilde{\alpha}_{mn}^{r} & = & \max\{0, a_{m,n-1} - a_{mn}\} \end{array} \tag{8}$$

Using this notation we can rewrite (6) as

$$DT_m = \sum_{n\in\mathcal{N}} \tilde{\alpha}_{mn}^{\ell} = \sum_{n\in\mathcal{N}} \tilde{\alpha}_{mn}^{r} \tag{9}$$

and the minimal decomposition time in the unconstrained case is

$$\max_{m\in\mathcal{M}} \sum_{n\in\mathcal{N}} \tilde{\alpha}_{mn}^{\ell} = \max_{m\in\mathcal{M}} \sum_{n\in\mathcal{N}} \tilde{\alpha}_{mn}^{r}. \tag{10}$$

In the next section we show that this representation of $\tilde{A}$ is essential to solve constrained decomposition time problems.

# 3 Constrained Decomposition Problem

In some applications, C1 matrices have to satisfy certain constraints, i.e., $\mathcal{K}' \subsetneq \mathcal{K}$. For example, in the radiotherapy application mentioned in Section 1, the mechanics of the multileaf collimator require that left and right leaves in adjacent rows must not overlap.

**Definition 3.1.** 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 $m = 2, ..., M$.

In this section we consider decomposition of $A$ into shape matrices to minimize decomposition time. We shall see that crucial intervals (Definition 2.1) and degenerate intervals are not sufficient to solve this problem. We may have to consider split crucial intervals, too. Let $\mathcal{K}'$ be an index set of all shape matrices.

There might not exist a decomposition of $A$ into shape matrices, obtained by using crucial intervals, which is an optimal solution of $\min\{DT(\alpha) : A = \sum_{k \in \mathcal{K}'} \alpha_k Y^k\}$.

For example

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

can be decomposed in the following way:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} [1,2) \\ [1,5) \\ [3,4) \\ [3,4) \end{pmatrix} + \begin{pmatrix} [7,8) \\ [5,8) \\ [6,6) \\ [5,6) \end{pmatrix}$$

with $DT(\alpha) = 2$. Note that crucial interval $[1,8)$ of the second row is split into two intervals and a degenerate interval is used in row three.

We have $\tilde{A} = \tilde{L} - \tilde{R}$:

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

Therefore, crucial intervals can be alternatively defined as $[\ell_m, r_m)$, where $\ell_m$ and $r_m$ correspond to the column indices of non–zero entries in row $m$ of $\tilde{L}$ and $\tilde{R}$, respectively.

A shape matrix decomposition of $A$ using crucial (and degenerate) intervals yields a decomposition time of at least 3. However, if we change the entries in the fifth column, second row and sixth column, third row of $\tilde{L}$ and $\tilde{R}$ simultaneously from 0 to 1 and use the alternative definition of crucial interval, we obtain the intervals $[1,5)$ and $[5,8)$ for the second row and the degenerate interval $[6,6)$ in the third row, which are used in the above shape matrix decomposition of $A$ with decomposition time 2 (which is the maximal row sum in both the modified $\tilde{L}$ and $\tilde{R}$). Note also that the degenerate interval in row three of $Y^2$ cannot be chosen arbitrarily. Because of $[5,8)$ in row two and $[5,6)$ in row four, only $[5,5)$ and $[6,6)$ are possible.

We proceed to show that this can always be done.

**Theorem 3.2.** *A has a decomposition with decomposition time $DT(\alpha)$ if and only if there exist $M \times (N+1)$ matrices $L = (\alpha^{\ell}_{mn})$ and $R = (\alpha^{r}_{mn})$ with non–negative elements such that*

$$
\begin{aligned}
L - R &= \tilde{A} && && (11)\\
\sum_{k=1}^{n} \alpha^{\ell}_{m-1,k} &\geq \sum_{k=1}^{n} \alpha^{r}_{mk} && \forall m \in \mathcal{M} \setminus \{1\},\ \forall n \in \mathcal{N} && (12)\\
\sum_{k=1}^{n} \alpha^{\ell}_{mk} &\geq \sum_{k=1}^{n} \alpha^{r}_{m-1,k} && \forall m \in \mathcal{M} \setminus \{1\},\ \forall n \in \mathcal{N} && (13)\\
DT(\alpha) = \sum_{k \in \mathcal{K}'} \alpha_k &= \sum_{n \in \mathcal{N}} \alpha^{\ell}_{pn} = \sum_{n \in \mathcal{N}} \alpha^{r}_{mn} && \forall p, m \in \mathcal{M} && (14)
\end{aligned}
$$

*Proof.* "$\Rightarrow$" Let a decomposition

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

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

$$
\begin{aligned}
\alpha^{\ell}_{mn} &= \sum_{\ell^k_m = n} \alpha_k,\\
\alpha^{r}_{mn} &= \sum_{r^k_m = n} \alpha_k.
\end{aligned}
$$

It is clear that (14) holds. From the elementwise presentation of the decomposition

$$
\begin{aligned}
a_{mn} &= \sum_{\ell^k_m < n < r^k_m} \alpha_k + \sum_{\substack{\ell^k_m = n\\ r^k_m > n}} \alpha_k,\\
a_{m,n-1} &= \sum_{\ell^k_m < n < r^k_m} \alpha_k + \sum_{\substack{\ell^k_m < n\\ r^k_m = n}} \alpha_k
\end{aligned}
$$

we get

$$
\begin{aligned}
a_{mn} - a_{m,n-1} &= \sum_{\substack{\ell^k_m = n\\ r^k_m > n}} \alpha_k - \sum_{\substack{\ell^k_m < n\\ r^k_m = n}} \alpha_k\\
&= \left( \sum_{\substack{\ell^k_m = n\\ r^k_m > n}} \alpha_k + \sum_{\substack{\ell^k_m = n\\ r^k_m = n}} \alpha_k \right) - \left( \sum_{\substack{\ell^k_m < n\\ r^k_m = n}} \alpha_k + \sum_{\substack{\ell^k_m = n\\ r^k_m = n}} \alpha_k \right)\\
&= \sum_{\ell^k_m = n} \alpha_k - \sum_{r^k_m = n} \alpha_k = \alpha^{\ell}_{mn} - \alpha^{r}_{mn}.
\end{aligned}
$$

Thus $L$ and $R$ satisfy (11).

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 $[\ell^k_m, r^k_m)$ where $r^k_m \leq n$. By definition of shape matrices each interval $[\ell^k_m, r^k_m)$ in a shape matrix $Y^k$ has a corresponding $[\ell^k_{m-1}, r^k_{m-1})$ in row $m-1$ such that

$$\ell^k_{m-1} \leq r^k_m.$$

Hence $\{Y^k \ : \ r^k_m \leq n\} \subseteq \{Y^k \ : \ \ell^k_{m-1} \leq n\}$ and we conclude

$$\sum_{r^k_m \leq n} \alpha_k \quad \leq \quad \sum_{\ell^k_{m-1} \leq n} \alpha_k$$

Consequently, we get

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

i.e., conditions (12) holds. Using a similar observation with $\ell^k_m \leq r^k_{m-1}$ we can derive (13).

"$\Leftarrow$" Let $L$ and $R$ be matrices $L$ and $R$ such that (11) – (14) hold. Let $\alpha^{\ell}_{m\ell_m}$ and $\alpha^{r}_{mr_m}, m \in \mathcal{M}$ be the first non–zero elements in the rows of matrices $L$ and $R$, respectively, i.e.,

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

From (11) we get for all $n \in \mathcal{N}$

$$\sum_{k=1}^{n} \alpha^{\ell}_{mk} - \sum_{k=1}^{n} \alpha^{r}_{mk} = a_{mn} \tag{15}$$

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

$$\sum_{k=1}^{n} \alpha^{\ell}_{mk} \geq \sum_{k=1}^{n} \alpha^{r}_{mk} \qquad \forall n \in \mathcal{N} \tag{16}$$

yields that

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

Moreover, from (12) and (13) 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

$$\alpha_1 = \min\left\{\alpha^{\ell}_{1\ell_1}, \ldots, \alpha^{\ell}_{M\ell_M}, \alpha^{r}_{1r_1}, \ldots, \alpha^{r}_{Mr_M}\right\}.$$

Replacing $\alpha^{\ell}_{m\ell_m}$ and $\alpha^{r}_{mr_m}$, $m = 1, \ldots, M$, by $\alpha^{\ell}_{m\ell_m} - \alpha_1$ and $\alpha^{r}_{mr_m} - \alpha_1$ in $L$ and $R$, respectively, we get matrices $L'$ and $R'$ which satisfy again (12), (13) and (16).

Thus by repeating the above procedure until the matrices $L$ and $R$ simultaneously become zero matrices, due to (14), we obtain a set of shape matrices $Y^1, \ldots, Y^k$ with corresponding $\alpha_1, \ldots, \alpha_k$.

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

$$\begin{aligned}\sum_{k=1}^{n} \alpha^{\ell}_{mk} &= \sum_{\ell^k_m \le n} \alpha_k, \\ \sum_{k=1}^{n} \alpha^{r}_{mk} &= \sum_{r^k_m \le n} \alpha_k.\end{aligned}$$

On the other hand,

$$\sum_{\ell^k_m \le n < r^k_m} \alpha_k = \sum_{\ell^k_m \le n} \alpha_k - \sum_{r^k_m \le n} \alpha_k = \sum_{k=1}^{n} \alpha^{\ell}_{mk} - \sum_{k=1}^{n} \alpha^{r}_{mk}.$$

Thus due to (15)

$$\sum_{\ell^k_m \le n < r^k_m} \alpha_k = a_{mn},$$

i.e.

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

□

According to Theorem 3.2 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. The following observation helps us to reduce the complexity of the problem. Using the denotation of (8), introduced at the end of Section 2,

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

we get for matrices $L$ and $R$, which satisfy (11) – (13),

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

and

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

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

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

$$\begin{aligned}\alpha^{\ell}_{mn} &= \tilde{\alpha}^{\ell}_{mn} + w_{mn} \\ \alpha^{r}_{mn} &= \tilde{\alpha}^{r}_{mn} + w_{mn}\end{aligned} \tag{17}$$

where $w_{mn} \ge 0$ and integer.

According to (14) the total decomposition time is, in terms of $L$ and $R$,

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

where $m$ is the index of any row of $A$. Therefore, we can use theorem 3.2 to formulate the decomposition time problem as the following integer linear programming problem (DT–IP)

$$\begin{array}{llcll}
\min & DT(\alpha) & & & \\
\text{s.t.} & DT_m + \sum_{k=1}^{N+1} w_{mk} & = & DT(\alpha) & \forall m \in \mathcal{M} \quad (18) \\
\sum_{k=1}^{n} \tilde{\alpha}^{\ell}_{m-1,k} + \sum_{k=1}^{n} w_{m-1,k} & & \geq & \sum_{k=1}^{n} \tilde{\alpha}^{r}_{mk} + \sum_{k=1}^{n} w_{mk} & \forall n \in \mathcal{N},\ \forall m \in \mathcal{M} \setminus \{1\} \quad (19) \\
& \sum_{k=1}^{n} \tilde{\alpha}^{\ell}_{mk} + \sum_{k=1}^{n} w_{mk} & \geq & \sum_{k=1}^{n} \tilde{\alpha}^{r}_{m-1,k} + \sum_{k=1}^{n} w_{m-1,k} & \forall n \in \mathcal{N},\ \forall m \in \mathcal{M} \setminus \{1\} \quad (20) \\
& w_{mn} & \geq & 0 \quad \text{integer} & \forall m \in \mathcal{M},\ \forall n \in \mathcal{N}
\end{array}$$

Note that the formulation of (DT–IP) is redundant since (18) follows from (19) and (20) with $n = N + 1$ and can be dropped. The minimization of $DT(\alpha)$ is then equivalent to minimizing $\sum_{k=1}^{N+1} w_{mk}$ for any choice of $m \in \mathcal{M}$, i.e. (DT–IP) is equivalent to, e.g.,

$$\left\{ \min \sum_{k=1}^{N+1} w_{mk} :\ (19), (20), w_{mn} \geq 0,\ \text{integer} \right\}.$$

In the following 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 3.3.** *Let $W = (w_{mn})$ be a feasible solution of (DT–IP). If for any column $p$, $w_p = (w_{1p}, w_{2p}, \ldots, w_{Mp})^T$, there exists $\bar{w} = (\bar{w}_1, \ldots, \bar{w}_M)^T \geq 0$ such that $w_p \geq \bar{w}$ and*

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

*for all $m = 2, \ldots, M$ 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. □

Based on Lemma 3.3 , we solve (DT–IP) recursively by solving a sequence of multiobjective integer programs $(SP_n)$, $n = 1, \ldots, N + 1$, in which the input data is defined by the output of $(SP_k)$, $k < n$. $(SP_n)$ is as follows.

$$
\begin{array}{llcll}
\min & \begin{pmatrix} w_{1n} \\ w_{2n} \\ \vdots \\ w_{Mn} \end{pmatrix} & & & \\
\text{s.t.} & DTL_{m-1}^n + w_{m-1,n} & \geq & DTR_m^n + w_{mn} & \forall m \in \mathcal{M} \setminus \{1\} \quad (21) \\
& DTL_m^n + w_{mn} & \geq & DTR_{m-1}^n + w_{m-1,n} & \forall\, m \in \mathcal{M} \setminus \{1\} \quad (22) \\
& w_{mn} & \geq & 0 \ \text{ integer} & \forall m \in \mathcal{M},\, \forall n \in \mathcal{N}
\end{array}
$$

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 $(w_{1k}^*, w_{2k}^*, \ldots, w_{mk}^*)^T$ is the optimal solution of (SP$_k$), $k < n$.

Due to (16) and (17) we get

$$
DTL_m^n \geq DTR_m^n \quad \forall m \in \mathcal{M} \tag{23}
$$

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 3.4.** *$(SP_n)$ has a unique Pareto optimal solution.*

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

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

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

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

$$\begin{array}{rcccl} DTL^n_{m-1} + \bar{w}_{m-1,n} & \geq & DTR^n_m + \bar{w}_{mn} & \geq & DTR^n_m + \hat{w}_{mn}, \\ DTL^n_m + \hat{w}_{mn} & \geq & DTR^n_{m-1} + \hat{w}_{m-1,n} & \geq & DTR^n_{m-1} + \bar{w}_{m-1,n}, \end{array}$$

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

**Algorithm 3.5 (Minimum C1 Decomposition Time into Shape Matrices).**

| | |
|---|---|
| **Input:** | *Matrix* $A$ |
| **Output:** | *Decomposition of* $A$ *into shape matrices with* $\min DT(\alpha)$ |
| *(1.)* | *Compute* $\tilde{\alpha}^\ell_{mn}, \tilde{\alpha}^r_{mn}$, $\forall\, m, n$. |
| *(2.)* | *For* $n = 1$ *to* $N + 1$<br>    *Solve* $(SP_n)$ *( with Algorithm 3.6)* |
| *(3.)* | *Compute matrices* $L$ *and* $R$; *and* $DT(\alpha)$ |
| *(4.)* | *Set* $k := 0$ |
| *(5.)* | *While* $DT(\alpha) \neq 0$ *do*<br>  *Consider leftmost non-zero elements*<br>    $\alpha^\ell_{m\ell_m}$ *and* $\alpha^r_{mr_m}$, $m = 1, \ldots, M$, *in each row of* $L$ *and* $R$<br>  $k := k + 1$<br>  *Extract shape matrix*<br>    $Y^k = Y([\ell_m, r_m))_{m\in\mathcal{M}}$ *with*<br>    $\alpha_k = \min\{\alpha^\ell_{1\ell_1}, \ldots, \alpha^\ell_{M\ell_M}, \alpha^r_{1r_1}, \ldots, \alpha^r_{Mr_M}\}$<br>  *Set* $DT(\alpha) := DT(\alpha) - \alpha_k$<br>  *Update* $L$ *and* $R$.<br>*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 3.6 (Solving $(SP_n)$).**

| | |
|---|---|
| **Input**: | $DTL^n_m, DTR^n_m, \quad \forall m = 1, \ldots, M.$ |
| **Output**: | $w^*_{mn}, \quad \forall m = 1, \ldots, M$ |
| *(1.)* | $w_{mn} := 0, \quad \forall m = 1, \ldots, M$ |
| *(2.)* | *For* $m = 2$ *to* $M$ *do*<br>  *if* $DTL^n_m + w_{mn} \leq DTR^n_{m-1} + w_{m-1,n}$<br>    *then* $w_{mn} := DTR^n_{m-1} - DTL^n_m + w_{m-1,n}$<br>    *else* $A(m)$<br>*end for* |

*Function A(p)*
*if* $DTL^n_{p-1} + w_{p-1,n} < DTR^n_p + w_{p,n}$
  *then* $w_{p-1,n} := DTR^n_p - DTL^n_{p-1} + w_{pn}$
  *if* $p \geq 3$
    *then* $p := p - 1$
    $A(p)$
  *end if*
*end if*
*end Function*

**Theorem 3.7.** *Algorithm 3.6 finds the optimal solution of* $(SP_n)$ *in* $\mathcal{O}(M^2)$ *time. Algorithm 3.5 solves (DT–IP) in* $\mathcal{O}(NM^2)$ *time.*

*Proof.* Obviously, the time complexity of Algorithm 3.6 is $\mathcal{O}(M^2)$. If we can prove that Algorithm 3.6 solves $(SP_n)$ to optimality, the time complexity of Algorithm 3.5 is $\mathcal{O}(NM^2)$. Hence the validity of Algorithm 3.6 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

$$DTL^n_1 + w_{1n} \geq DTR^n_2 + w_{2n}, \tag{24}$$
$$DTL^n_2 + w_{2n} \geq DTR^n_1 + w_{1n} \tag{25}$$

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: $DTL^n_2 \leq DTR^n_1$. Then $w_{2n} = DTR^n_1 - DTL^n_2$ from step (2.) is a lower bound by (24) and (25); and satisfies $DTR^n_2 + w_{2n} = DTR^n_2 + DTR^n_1 - DTL^n_2 \leq DTR^n_2 + DTR^n_1 - DTR^n_2 = DTR^n_1 \leq DTL^n_1$, due to (23). Hence $(w_{1n} = 0, w_{2n})$ is feasible.
- Case 2: $DTL^n_2 > DTR^n_1$. Then $w_{2n} = 0$ is the lower bound due to (24) and (25). The lower bound for $w_{2n}$ is tightened using Function A(2)
  - If $DTL^n_1 < DTR^n_2$ then using (23) $w_{1n} = DTR^n_2 - DTL^n_1$ satisfies $DTR^n_1 + w_{1n} = DTR^n_1 + DTR^n_2 - DTL^n_1 \leq DTR^n_1 + DTR^n_2 - DTR^n_1 = DTR^n_2 \leq DTL^n_2$ such that $(w_{1n}, w_{2n} = 0)$ is feasible.
  - If $DTL^n_1 \geq DTR^n_2$ then $(w_{1n} = 0, w_{2n} = 0)$ is feasible.

$m < M$: Assume that Algorithm 3.6 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 $1, \ldots, M - 1$. This solution can serve as a lower bound for $w_{mn}$, $m = 1, \ldots, M - 1$ of problem $(SP_n)$ defined for rows $1, \ldots, M$. Now we tighten this bound with respect to constraints

$$DTL^n_{M-1} + w_{M-1,n} \geq DTR^n_M + w_{Mn}, \tag{26}$$
$$DTL^n_M + w_{Mn} \geq DTR^n_{M-1} + w_{M-1,n} \tag{27}$$

which contain variable $w_{Mn}$.

- Case 1: If $DTL_M^n \le DTR_{M-1}^n + w_{M-1,n}$ then the lower bound for $w_{Mn}$ is $w_{Mn} = DTR_{M-1}^n + w_{M-1} - DTL_M^n$, which satisfies both inequalities since $DTR_M^n + DTR_{M-1}^n + w_{M-1,n} - DTL_M^n \le DTR_M^n + DTR_{M-1}^n + w_{M-1,n} - DTR_M^n = DTR_{M-1}^n + w_{M-1,n} \le DTL_{M-1}^n + w_{M-1,n}$ due to (23).
- Case 2: If $DTL_M^n > DTR_{M-1}^n + w_{M-1,n}$ then $w_{Mn} = 0$
  - If the (26) 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 (27), holds by the choice of $w_{M-1,n}$ and (23). The second one is
  $$DTL_{M-2}^n + w_{M-2,n} \ge DTR_{M-1}^n + w_{M-1,n}.$$
  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 3.8.**

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

The corresponding matrices

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

of $A$ are defined according to (8). 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

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

which correspond to an optimal solution of the decomposition time problem with $DT(\alpha) = 10$. Extracting shape matrices, with respect to the most left 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 $\sum_{k=1}^{K_A} \alpha_k = 10 = DT$.

In the next section we need the following proposition to find some easily solvable instances of the decomposition cardinality problem.

**Proposition 3.9.** *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 Lemmas 2.2 and 2.3 and Algorithm 2.5. For the constrained case, observe that if we neglect the integrality of the coefficients $\alpha_k$, $k \in \mathcal{K}'$, then the statement follows from the (DT–IP) formulation with respect to A and B. On the other hand, Algorithms 3.5 and 3.6 yield an integer solution only due to integrality of the input matrix. This completes the proof. $\square$

# 4 Decomposition Cardinality is NP-hard

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) for matrices with at least two rows using a reduction from subset sum. In the following we will strengthen his result.

**Theorem 4.1.** *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 (DC) to the following well-known strongly NP-complete problem (see Garey and Johnson (1979) ).

**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 $q = 1, ..., Q$?

We define

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

**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).

"$\Rightarrow$" By the definition of $a_n$ it 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 $\alpha_q,\ q = 1, ..., 3Q$, such that the sum of the interval lengths is maximized. We derive the following properties.

1. For all $p, q \in \{1, \dots, 3Q\}$ $\ell_q \neq r_p$. Otherwise we can replace $I_p$ and $I_q$ by $I'_p := I_p \cup I_q$ with $\alpha'_p := \min\{\alpha_p, \alpha_q\}$ and $I'_q := I_q$ with $\alpha'_q := \alpha_q - \alpha'_p$ to get a C1 decomposition with larger interval lengths.
2. Without loss of generality $\ell_q = q$ for all $q = 1, ..., 3Q$. This follows since $a_1 < a_2 < ... < a_{3Q}$ and some interval has to start in $q$.
3. $r_q > 3Q$ for all $q = 1, ..., 3Q$. Otherwise, we have a contradiction to 1 and 2 with $l_p = r_q$ for some $p = 1, \dots, 3Q$.
4. $r_q \neq 3Q + 1$. Otherwise some $\ell_q = 3Q + 1$ would be needed since $a_{3Q} = a_{3Q+1}$. This would contradict 2.

Hence all intervals end in $r_q \in \{3Q+2, ..., 4Q+1\}$. 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.

□

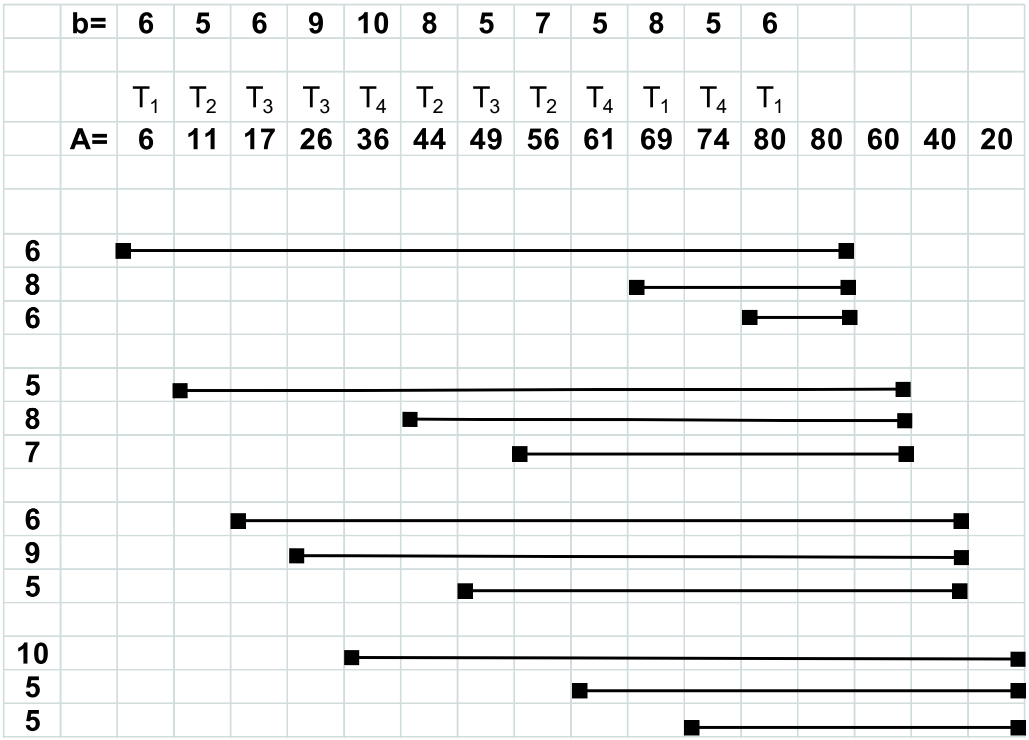



Figure 4: 3-PART $\propto$ DC with $B = 20, Q = 4$.

**Corollary 4.2.** *Even if $L$ and $R$ are matrices known to correspond to an optimal solution of the DC problem, the problem of finding that optimal decomposition (with respect to the DC objective) is strongly NP-hard.*

*Proof.* Follows from the proof of Theorem 4.1 where intervals of maximal lengths are used. Therefore, no intervals $[\ell_1, r_1)$ and $[\ell_2, r_2)$ with $r_1 = \ell_2$ exist implying that $W = 0$ and thus $L = \tilde{L}$ and $R = \tilde{R}$. □

In some cases DC, however, can be solved in polynomial time.

**Proposition 4.3.** *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 $\alpha_k$ is binary for all $k \in \mathcal{K}'$. Hence, if matrix $A$ is a binary matrix then we can use Algorithm 3.5 (for the unconstrained case Algorithm 2.5) 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 3.5 (for the unconstrained case Algorithm 2.5) 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 of $A$. Without loss of generality we can assume 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 $k = 1, \ldots, k_0$. Let $A'$ and $B'$ be matrices defined as

$$\begin{aligned} A' &:= A - p \sum_{k=k_0+1}^{K} Y^k = \sum_{k=1}^{k_0} \alpha_k Y^k \\ B' &:= B - \sum_{k=k_0+1}^{K} Y^k. \end{aligned}$$

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 3.9 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 $k = 1, \ldots, k_0$. 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. □

Next we develop heuristics for the DC problem. As we have seen in the proof of Theorem 3.2 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 $L$ and $R$ 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 matrices $L$ and $R$. If we use $\tilde{L}$ and $\tilde{R}$, (17) yields the following lower bound for the decomposition cardinality problem

$$DC(\alpha) \geq \left\{ \min k_0 : \ k_0 \geq \mid \{\alpha^\delta_{mn} : \alpha^\delta_{mn} \neq 0, \ n \in \mathcal{N}\} \text{ for all } m \in \mathcal{M} \text{ and } \delta \in \{\ell, r\} \right\}.$$

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 in 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 3.2 and consequently in the Algorithm 3.5 we used leftmost non-zero elements in the rows of $L$ and $R$, which preserve conditions (12), (13) and (16). 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

$$\bar{a}_{mn} = \sum_{k=1}^{n} \alpha^\ell_{mk} - \sum_{k=1}^{n} \alpha^r_{m+1,k} \tag{28}$$

$$\hat{a}_{mn} = \sum_{k=1}^{n} \alpha^\ell_{m+1,k} - \sum_{k=1}^{n} \alpha^r_{mk} \tag{29}$$

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

Then conditions (12), (13) and (16) for residual matrices of $L$, $R$ and $A$ can be written in terms of $\bar{A}$ and $\hat{A}$, respectively, as

$$\begin{array}{rcll} \bar{a}_{mn} & \geq & \alpha & \forall n:\ \ell_m \leq n < r_{m+1}, \quad \forall m \in \mathcal{M} \setminus \{M\} \\ \hat{a}_{mn} & \geq & \alpha & \forall n:\ \ell_{m+1} \leq n < r_m, \quad \forall m \in \mathcal{M} \setminus \{M\} \\ a_{mn} & \geq & \alpha & \forall n:\ \ell_m \leq n < r_m, \qquad \forall m \in \mathcal{M}, \end{array} \begin{array}{r} (30) \\ (31) \\ (32) \end{array}$$

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

$$\begin{array}{rcll} & \max & \alpha & \\ \text{s.t. } (30), (31), (32) & & & \\ \ell_m & \leq & r_{m+1} & \forall m \in \mathcal{M} \setminus \{M\} \quad (33) \\ \ell_{m+1} & \leq & r_m & \forall m \in \mathcal{M} \setminus \{M\} \quad (34) \\ \ell_m & \leq & r_m & \forall m \in \mathcal{M} \quad (35) \\ \alpha^{\ell}_{m\ell_m} & \geq & \alpha & \forall m \in \mathcal{M} \quad (36) \\ \alpha^{r}_{mr_m} & \geq & \alpha & \forall m \in \mathcal{M} \quad (37) \\ \ell_m,\ r_m & \in & \mathcal{N} & \forall m \in \mathcal{M} \end{array}$$

In Baatar and Hamacher (2003) CPLEX 7.0 was used to solve mixed integer formulation of ($\max -\alpha$). Since the computation time were prohibitively large we propose in the following a combinatorial approach, which produces objective values superior to these of Alfredo and Siochi (1999), Xia and Verhey (1999) and Bortfeld *et al.* (1994).

**Algorithm 4.4 (Greedy Approach to the Constrained Decomposition Cardinality Problem).**

| | |
|---|---|
| **Input:** | *Matrix $A$* |
| **Output:** | *Decomposition of $A$ into shape matrices* |
| *(1.)* | *Compute $DT(\alpha)$ and matrices $L$ and $R$ using Algorithm 3.6* |
| *(2.)* | *Compute $\bar{A}$, $\hat{A}$ according to (28) and (29)* |
| *(3.)* | *Initialize $\alpha := \min\{\max_{n\in\mathcal{N}}\{\alpha^{\delta}_{mn} :\ \alpha^{\delta}_{mn} < \alpha\} :\ m \in \mathcal{M},\ \delta \in \{\ell, r\}\}$* |
| | *Set $k := 0$* |
| *(4.)* | *While $DT(\alpha) \neq 0$ do* |
| *(4.1.)* | *If $\alpha \neq 1$* |
| | *then For $m = 1$ to $M$ do* |
| | $I_m := \{[p,q) : (32), (35) - (37)\}$ |
| | *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* |
| | $m := 1$ |
| | $[\ell_1, r_1) := lexmin\{[p,q) :\ [p,q) \in I_1\}$ |
| | *end If* |
| | *Remove all intervals $[p,q)$, from $I_{m+1}$, with $q < \ell_m$* |

*If* $I_{m+1} = \{\emptyset\}$ *then GO TO (4.9.)*
*Find 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*
*(32),(36) and (37) 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 most–left–non–zero elements of* $L$ *and* $R$*)*
*end If*
*(4.9.)* $\alpha := \alpha - 1$
*(4.10.)* *end while (4.)*
*End.*

---

Algorithm 4.4 considers iteratively all possible values of $\alpha$ in the while loop (4.). Whenever $\alpha = 1$, i.e., the maximal possible coefficient is one 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 (32), (35)–(37). In the while loop (4.3.) these sets are iteratively reduced with respect to conditions (30), (31), (33) and (34) 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 end 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 $\alpha$, $\alpha$ is updated in (4.9.) and the above procedure is repeated for the new value of $\alpha$.

**Example 4.5.** Consider the matrix $A$ given in Example 3.8. Using the matrices

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

found in this example, we can compute

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

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

$$\begin{aligned} I_1 &= \{[1,3),[1,4),[2,3),[2,4)\} \\ I_2 &= \{[1,2)\} \\ I_3 &= \{[1,2)\} \end{aligned}$$

The first elements of these sets satisfy conditions (30), (31), (33) and (34), i.e., we can extract a shape matrix

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

Updating interval sets, using conditions (32), (36) and (37), with respect to residual matrices

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

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 (33) and (34) with $[4,4) \in I_2$. The remaining intervals form the shape matrix

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

which satisfy (30) and (31). Repeating the above procedure we get

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

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

So we have a alternative decomposition of A with smaller number of shape matrices compared with the decomposition in Example 3.8.

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

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

where $W$ is any matrix with positive integer entries such that

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

for all $m \in \mathcal{M}$. For instance, we can choose $W$ such that each row $m$ of $W$ has not 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 $DT - DT_m$.

Problem ($\max - \alpha$) now becomes

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad (32),(35),(36),(37) & \\ \ell_m,\ r_m \quad & \in \quad \mathcal{N} \quad \forall m \in \mathcal{M} \end{aligned}$$

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

**Algorithm 4.6 (Greedy Approach to Unconstrained Decomposition Cardinality Problem).**

**Input:** *Matrix A*
**Output:** *Decomposition of A into C1 matrices*
*(1.)* Compute $DT$, $DT_m$, $m \in \mathcal{M}$
*(1.)* Compute matrices $L$ and $R$
*(2.)* Initialize $\alpha := \min\{\max_{n\in\mathcal{N}}\{\alpha^{\delta}_{mn} : \alpha^{\delta}_{mn} < \alpha\} : \forall m \in \mathcal{M},\ \delta \in \{\ell, r\}\}$
*(3.)* Set $k := 0$
*(4.)* While $DT \neq 0$ do
    For $m = 1$ to $M$ do
        $I_m := \{[p,q) : (32),(35)-(37)\}$
        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
                    (32),(36) and (37) 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.

# References


Ahuja, R. and Hamacher, H. (2004). Linear time network flow algorithm to minimize beam-on-time for unconstrained multileaf collimator problems in cancer radiation therapy. Revised version under review in Networks.

Alfredo, R. and Siochi, C. (1999). Minimizing static intensity modulation delivery time using an intensity solid paradigm. *International Journal of Radiation Oncology, Biology, Physics*, **43**(3), 671–680.

Baatar, D. and Hamacher, H. (2003). New LP model for multileaf collimators in radiation therapy planning. In *Proceedings of the Operations Research Peripatetic Postgraduate Programme Conference ORP*$^3$*, Lambrecht, Germany*, pages 11–29.

Boland, N., Hamacher, H., and Lenzen, F. (2003). Minimizing beam-on time in cancer radiation treatment using multileaf collimators. *Networks*, **43**(4), 226–240.

Booth, K. and Lucker, G. (1976). Testing for the consecutive ones property, interval graphs and graph planarity using PQ–tree algorithms. *Journal of Computer and System Science*, **13**, 335–379.

Bortfeld, T., Boyer, A., Kahler, D., and Waldron, T. (1994). X-ray field componsation with multileaf collimators. *International Journal of Radiation Oncology, Biology, Physics*, **28**(3), 723–730.

Burkard, R. (2002). Open Problem Session, Oberwolfach Conference on Combinatorial Optimization, November 24-29, 2002.

Engel, K. (2003). A new algorithm for optimal MLC field segmentation. Technical report, Department of Mathematics, University of Rostock.

Garey, M. and Johnson, D. (1979). *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., San Francisco, CA.

Hamacher, H. and Küfer, K.-H. (2002). Inverse radiation therapy planing – A multiple objective optimization approach. *Discrete Applied Mathematics*, **118**, 145–161.

Hamacher, H., Liebers, A., Schöbel, A., Wagner, D., and Wagner, F. (2001). Locating new stops in a railway network. *Electronic Notes in Theoretical Computer Science*, **50**(1). www.elsevier.nl/entcs/volume50.html.

Kalinowski, T. (2003). *An algorithm for optimal MLC field segmentation with interleaf collision constraints*. Master's thesis, University of Rostock, Germany.

Nemhauser, G. and Wolsey, L. (1988). *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York.

Ruf, N. and Schöbel, A. (2003). Set covering problems with the almost consecutive ones property. Technical report, Department of Mathematics, University of Kaiserslautern.

Schöbel, A., Hamacher, H., Liebers, A., and Wagner, D. (2002). The continuous stop location problem in public transportation networks. Technical Report 81, Department of Mathematics, University of Kaiserslautern. Submitted to Transportation Science.

Xia, P. and Verhey, L. (1999). Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple segments. *Medical Physics*, **25**, 1424–1434.