Constrained Decompositions of Integer Matrices and their Applications to Intensity Modulated Radiation Therapy*

Céline Engelbeen† Samuel Fiorini‡

November 10, 2008

Abstract

We consider combinatorial optimization problems arising in radiation therapy. Given a matrix I with non-negative integer entries, we seek a decomposition of I as a weighted sum of binary matrices having the consecutive ones property, such that the total sum of the coefficients is minimized. The coefficients are restricted to be non-negative integers. Here, we investigate variants of the problem with additional constraints on the matrices used in the decomposition. Constraints appearing in the application include the interleaf motion and interleaf distance constraints. The former constraint was previously studied by Baatar *et al.* (*Discrete Appl. Math., 2005*) and Kalinowski (*Discrete Appl. Math., 2005*). The latter constraint was independently considered by Kumar *et al.* (*working paper, 2007*) in the case where coefficients of the decomposition are not restricted to be integers. For both constraints, we prove that finding an optimal decomposition reduces to finding a maximum value potential in an auxiliary network with integer arc lengths and no negative length cycle. This allows us to simplify and unify the previous approaches. Moreover, we give an $O(MN + KM)$ algorithm to solve the problem under the interleaf distance constraint, where M and N respectively denote the number of rows and columns of the matrix I and K is the number of matrices used in the decomposition. We also give an $O(MN \log M + KM)$ algorithm for solving the problem under the interleaf motion constraint and hence improve on previous results. Finally, we show the problem can still be solved in $O(MN \log M + KM)$ time when both constraints are considered simultaneously.

Keywords: Decomposition of integer matrices, consecutive ones property, multileaf collimator, radiation therapy.

*This research is supported by an "Actions de Recherche Concertées" (ARC) projet of the "Communauté française de Belgique". Céline Engelbeen is a research fellow of the "Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture" (FRIA).

†Department of Mathematics, Université Libre de Bruxelles CP 216, Boulevard du Triomphe, 1050 Brussels, Belgium. Email: `cengelbe@ulb.ac.be`.

‡Department of Mathematics, Université Libre de Bruxelles CP 216, Boulevard du Triomphe, 1050 Brussels, Belgium. Email: `sfiorini@ulb.ac.be`.

1 Introduction

When a cancer is diagnosed, a physician can prescribe radiation therapy sessions. The aim of these sessions is to destroy the tumor by exposing it to radiation while preserving the organs located in the radiation field, called *organs at risk*. Note that in order to achieve this goal it is necessary to use different radiation angles. Nowadays, in large hospitals, radiation is delivered by a linear accelerator equipped with a multileaf collimator (*MLC*) (see Figure 1).

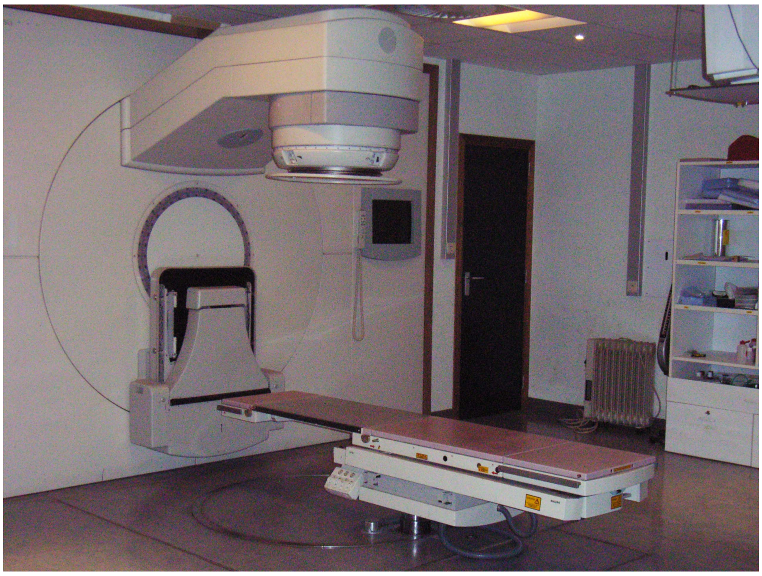

Figure 1: A complete treatment unit (gantry with a linear accelerator, MLC and couch) at Saint-Luc Hospital (Brussels, Belgium).

Designing a radiation plan is a complex optimization problem. Currently, the problem is solved in three distinct phases. The input to the problem basically consists of a 3-dimensional model of the patient's body, usually obtained by computerized tomography. The physician identifies the tumor and the organs at risk within the 3-dimensional model and prescribes a lower bound on the radiation dose for the tumor and an upper bound on the radiation dose for each organ at risk. The radiation plan is then designed as follows:

The first phase is to choose the different radiation angles.

In the second phase an intensity function is constructed for each radiation angle; see, for example, [9]. This function is encoded as an integer matrix I of dimension $M \times N$. Each entry of the matrix corresponds to an elementary part of the radiation beam, called a *bixel* (or *beamlet*). The value of an entry gives the required intensity, that is the exposure time, for the corresponding bixel.

Without MLC, linear accelerators produce homogeneous radiation fields. That means that linear accelerator delivers radiation with the same intensity across the whole beam. The MLC consists of pairs of metallic leaves (between 40 and 60 pairs) located between the radiation source and the patient. The role of the MLC is to block part of the radiation (see Figure 2). In *segmented* (or *step-and-shoot*) intensity modulated radiation therapy, the leaves do not move when the radiation beam is on. There also exists a dynamic mode. In this mode, the leaves continuously move during the radiation. We do not consider this mode in the present paper.

The third phase in the design of a radiation plan is to describe how radiation is modulated by the leaves of the MLC. Mathematically, this phase consists of modulating the intensity matrix I as a weighted sum of binary matrices called "segments". Entries equal to 1 indicate that the corresponding bixel is on and entries equal to 0 indicate that the radiation in the corresponding bixel is blocked by a leaf. Each segment has to satisfy the *consecutive ones property*, that is, has to have its ones grouped in a single block within each row (see Figure 2). Moreover, the coefficients of the decomposition are usually constrained to be integers.

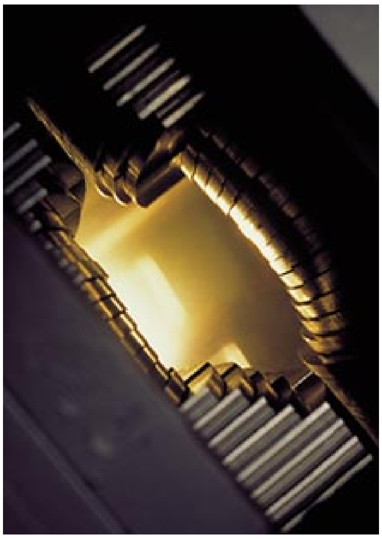

Figure 2: The multileaf collimator (MLC).

Consider for example the following intensity matrix:

$$I = \begin{pmatrix} 5 & 3 & 5 & 1 \\ 3 & 1 & 3 & 3 \\ 5 & 1 & 1 & 3 \\ 5 & 3 & 3 & 1 \end{pmatrix}.$$

A possible decomposition of I into segments is as follows:

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

This decomposition describes the way the leaves of the MLC are placed to modulate the intensity matrix. First, the leaves take the shape of first segment (all the leaves are open in this case) and the patient is irradiated for 1 monitor unit (radiation passes thus through each bixel). Next, the leaves move and take the shape of the second segment and the patient is irradiated for 2 monitor units, and similarly for the last two segments. The sum of the coefficients of the segments exactly corresponds to the time during which the patient is irradiated. We call this time the *beam-on time*. The problem we consider in the present paper is to find a decomposition that minimizes the beam-on time. We refer to the problem as the *minimum beam-on-time problem* (shortly, *BOT*). We consider several variants of this problem where further constraints are imposed on the segments. Any such problem can be written as an integer program in the following way.

$$\begin{aligned} \text{(BOT)} \quad \min \quad & \sum_{S \in \mathcal{S}} \alpha_S \\ \text{s.t.} \quad & \sum_{S \in \mathcal{S}} \alpha_S S = I \\ & \alpha_S \in \mathbb{Z}_+ \quad \forall S \in \mathcal{S}. \end{aligned}$$

In the unconstrained case, $\mathcal{S}$ is the whole set of $M \times N$ binary matrices satisfying the consecutive ones property. In the presence of a constraint, $\mathcal{S}$ contains fewer matrices. Note that the above integer programming formulation has $(\binom{N+1}{2} + 1)^M$ variables and MN equality constraints.

Efficient methods for solving BOT in the unconstrained case have been proposed by several authors, for example, Xia and Verhey [17], Siochi [16], Ahuja and Hamacher [1], Baatar, Hamacher, Ehrgott and Woeginger [3], Engel [7], Kalinowski [10] and Kamath, Sahni, Li, Palta and Ranka [12]. The given references do not only treat the unconstrained case. The relation between several approaches to the problem given in [3, 10, 12] is explained in [11]. In the next paragraphs, we recall the approach of Baatar *et al.* [3].

Consider some segment $S = (s_{mn})$. Let ℓ_m and r_m respectively denote the position of the left and the right leaves in the m-th row of S (see Figure 3 for an example). Formally, we have

$$\begin{aligned} \ell_m &:= \min\{n : s_{mn} = 1\}, \quad \text{and} \\ r_m &:= 1 + \max\{n : s_{mn} = 1\}. \end{aligned}$$

If the m-th row of S is zero (that is, the radiation is blocked on row m), we let ℓ_m and r_m be any integers between 1 and $N + 1$ such that $\ell_m = r_m$. So in this case, the positions of the leaves are not entirely determined by the segment S. For the unconstrained version of BOT this is not a problem. But for the constrained versions we consider, every leaf position has to be specified. Consequently, we sometimes denote by $([\ell_m, r_m])_{m=1,\ldots,M}$ the segment whose m-th left and right leaves are respectively in positions ℓ_m and r_m, for $m = 1, \ldots, M$.

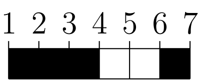

Figure 3: A segment with one row, left leaf in position 4 and right leaf in position 6.

Let $I = (i_{mn})$ denote a non-negative integer matrix of size $M \times N$. For the sake of conciseness, we call a matrix a *segment* if it is binary and satisfies the consecutive ones property. We would like to decompose I as a weighted sum of segments with integer coefficients such that the sum of the coefficients is minimized. Let $\Delta = \Delta(I) = (\delta_{mn})$ be the $M \times (N + 1)$ matrix defined as

$$\delta_{mn} := i_{mn} - i_{m,n-1} \qquad \forall m \in \{1, \ldots, M\},\ \forall n \in \{1, \ldots, N + 1\},$$

where we let $i_{m,0} = i_{m,N+1} := 0$. If $\delta_{mn} > 0$ we know that $i_{mn} > i_{m,n-1}$. Therefore, for at least $i_{mn} - i_{m,n-1} = \delta_{mn}$ time units, the radiation has to pass through bixel (m, n) and not through bixel

$(m, n-1)$. To achieve this the left leaf in the m-th row has to be placed in position n for at least δ_{mn} time units. So, the positive entries of the matrix Δ give a lower bound on the time during which the left leaves have to be in a certain position. Similarly, if $\delta_{mn} < 0$, then $i_{mn} < i_{m,n-1}$. The radiation has to pass through bixel $(m, n-1)$ and not through bixel (m, n) for at least $-\delta_{mn}$ time units. We therefore have to place the right leaf of the m-th row in position n for at least $-\delta_{mn}$ time units. So, the negative entries of the matrix Δ give a lower bound on the time during which the right leaves have to be in a certain position. Let $\Delta^+ = (\delta^+_{mn})$ and $\Delta^- = (\delta^-_{mn})$ be the matrices of size $M \times (N+1)$ defined by:

$$\delta^+_{mn} := \max\{0, \delta_{mn}\} \quad \text{and} \quad \delta^-_{mn} := \max\{0, -\delta_{mn}\}.$$

The above discussion immediately leads to the following lower bound on the total beam-on time.

Lemma 1.1. *Letting OPT denote the beam-on time of an optimal solution to the unconstrained minimum beam-on-time problem, we have $OPT \geq \max\{T_m : m = 1, \ldots, M\}$ where*

$$T_m := \sum_{n=1}^{N+1} \delta^+_{mn} = \sum_{n=1}^{N+1} \delta^-_{mn} \tag{1}$$

is the unavoidable beam-on-time *for the m-th row.*

As observed for example by Baatar *et al.* [3] and Bortfeld, Kahler, Waldron and Boyer [6], it turns out that the bound given by Lemma 1.1 is exact. Other authors, for example, Engel [7] and Kalinowski [10], have proved the same result. Since all the rows of I are independent, each row can be decomposed separately. For the time being, we consider the m-th row of I for some fixed m and seek a decomposition of this row with beam-on-time T_m. We may regard the matrix Δ^+ (resp. Δ^-) as a multiset giving the number of *unavoidable* left (resp. right) leaf positions; see Figure 4 for an illustration. Note that this naturally determines an ordering of the unavoidable leaf positions.

$\Delta^+ = (5 \quad 0 \quad 0 \quad 2 \quad 0)$, $\qquad \Delta^- = (0 \quad 4 \quad 0 \quad 0 \quad 3)$

Figure 4: Left and right leaf positions for the row $(5 \quad 1 \quad 1 \quad 3)$.

The idea of the algorithm of Baatar *et al.* [3] is to match each unavoidable left leaf position with some unavoidable right leaf position in order to obtain a decomposition of the given intensity matrix I. We point out that it is important to carefully pick the matching otherwise one might obtain a matrix different from I. A matching that always works is constructed by iteratively associating the first unmatched unavoidable left leaf position with the first unmatched unavoidable right leaf position. Actually, any matching between the left and the right leaf positions works, as long as we always have $\ell_m \leq r_m$.

The resulting algorithm can be extended in two ways. First, by considering all rows simultaneously and independently, the algorithm can produce a decomposition of the whole intensity matrix I. Second, for the constrained case it is necessary to replace the pair of matrices Δ^+ and Δ^- by a more general pair of non-negative integer matrices D^+ and D^- such that $D^+ - D^- = \Delta$. After implementing these two extensions, we obtain the *standard decomposition algorithm* (see Algorithm 1 for a formal description). To solve the beam-on-time problem in the unconstrained case, we take $D^+ = \Delta^+$ and $D^- = \Delta^-$.

Algorithm 1 Standard decomposition algorithm.

Input: Non-negative integer matrices D^+ and D^- of size $M \times (N+1)$ with $D^+ - D^- = \Delta$ representing leaf positions, given as multisets $D^+_1, \ldots, D^+_M$ and $D^-_1, \ldots, D^-_M$.
Output: A decomposition of I with the prescribed leaf positions.

while there exists some $m \in \{1, \ldots, M\}$ such that $D^+_m \neq \varnothing$ **do**
  for all $m \in \{1, \ldots, M\}$ such that $D^+_m \neq \varnothing$ **do**
    $\ell_m \leftarrow \min D^+_m$; $r_m \leftarrow \min D^-_m$
    $\alpha_m \leftarrow \min\{\text{multiplicity of } \ell_m \text{ in } D^+_m, \text{ multiplicity of } r_m \text{ in } D^-_m\}$
  end for
  for all $m \in \{1, \ldots, M\}$ such that $D^+_m = \varnothing$ **do**
    $\ell_m \leftarrow N+1$; $r_m \leftarrow N+1$
    $\alpha_m \leftarrow +\infty$
  end for
  $\alpha \leftarrow \min\{\alpha_1, \ldots, \alpha_M\}$
  for all $m \in \{1, \ldots, M\}$ such that $D^+_m \neq \varnothing$ **do**
    remove α copies of ℓ_m (resp. r_m) from D^+_m (resp. D^-_m)
  end for
  output α times the segment $([\ell_m, r_m))_{m=1,\ldots,M}$
end while

We now sketch a proof of the correctness of the standard decomposition algorithm. The key property we need to prove the correctness of the algorithm is the equation $D^+ - D^- = \Delta$. Letting $D^+ := (d^+_{mn})$ and $D^- := (d^-_{mn})$, the latter equation is equivalent to:

$$\sum_{j=1}^{n} d^+_{mj} - \sum_{j=1}^{n} d^-_{mj} = i_{mn} \qquad \forall m \in \{1, \ldots, M\},\ \forall n \in \{1, \ldots, N+1\}. \tag{2}$$

We claim that as soon Equation (2) is satisfied, Algorithm 1 outputs a decomposition of I whose beam-on-time equals the size of the largest multiset D^+_m or D^-_m.

Recall that we let $i_{m,N+1} = 0$ by convention. Thus Equation (2) for $j = N+1$ implies that the multisets D^+_m and D^-_m have the same size. So at the end of the algorithm both D^+_m and D^-_m are empty, for all m. Equation (2) for the other values of j implies that we always have $\ell_m \leq r_m$. So the algorithm outputs valid segments. The sum of all the coefficients α output by the algorithm is clearly the size of the largest multiset D^+_m or D^-_m. It remains to verify that the sum of the decomposition output by the algorithm is I. Fix some $m \in \{1, \ldots, M\}$ and some $n \in \{1, \ldots, N\}$.

The bixel (m, n) is irradiated exactly when the position of the left leaf is less than or equal to n and the position of the right leaf is strictly greater than n. Since the algorithm associates the first unmatched left leaf position with the first unmatched right leaf position, the bixel (m, n) is irradiated during $\sum_{j=1}^{n} d_{mj}^{+} - \sum_{j=1}^{n} d_{mj}^{-} = i_{mn}$ times unit, as required.

As pointed out before, in the unconstrained case we take $D^+ = \Delta^+$ and $D^- = \Delta^-$. In this case we have $D^+ - D^- = \Delta$ and thus Equation (2) is satisfied. By what precedes, Algorithm 1 finds a decomposition of I whose beam-on-time equals the lower bound given by Lemma 1.1, that is, the maximum unavoidable beam-on-time of a row. Therefore, the decomposition of I output by the standard decomposition algorithm is optimal. The complexity of the algorithm is $O(KM)$ where K is the number of segments output. Note that we have $K \leq 2MN$ (even $K \leq MN$ in the unconstrained case), so in particular the standard decomposition algorithm runs in time $O(M^2N)$. Because the matrices Δ, Δ^+ and Δ^- and the corresponding multisets can be computed from I in $O(MN)$ time the unconstrained minimum beam-on-time problem can be solved in $O(MN+KM)$ time. This concludes our description of the algorithm of Baatar *et al.* for the unconstrained case.

We now describe the two constraints studied here. These are all motivated by mechanical constraints of certain types of multileaf collimators. The *interleaf distance constraint* asks that the distance between the positions of two left (resp. right) leaves never exceeds a given constant c. In other words, a segment $S = ([\ell_m, r_m))_{m=1,\dots,M}$ satisfies the interleaf distance constraint if and only if we have

$$|\ell_m - \ell_{m'}| \leq c \quad \forall m, m' \in \{1, \dots, M\} \tag{3}$$

$$|r_m - r_{m'}| \leq c \quad \forall m, m' \in \{1, \dots, M\}. \tag{4}$$

This constraint appears briefly in Kumar, Ahuja and Kamath [13] under the name *maximum leaf spread constraint* in the case the coefficients of the decomposition are not restricted to be integers. Our main result is that BOT under the interleaf distance constraint is a polynomial problem and can be solved in time $O(MN + KM)$, that is, essentially as fast as the unconstrained problem.

The *interleaf motion constraint* (also known as the *interleaf collision constraint* or the *interdigitation constraint*) forbids the left leaf of some row to overlap the right leaf of an adjacent row. A segment $S = ([\ell_m, r_m))_{m=1,\dots,M}$ satisfies the interleaf motion constraint if and only if we have

$$\ell_m \leq r_{m+1} \quad \forall m \in \{1, \dots, M-1\}$$

$$\ell_m \leq r_{m-1} \quad \forall m \in \{2, \dots, M\}.$$

Baatar *et al.* [3] and Kalinowski [10] independently found algorithms to solve BOT under the interleaf motion constraint. Baatar *et al.*'s algorithm has complexity $O(M^2N+KM)$. Kalinowski's algorithm has complexity $O(OPT\, M^2N)$, where OPT denotes the optimal beam-on-time. We propose a new and faster algorithm of complexity $O(MN \log M + KM)$. Moreover, we show that we can solve BOT under both constraints within the same time bound.

Before giving the outline of the paper in the next paragraph we briefly explain how the constrained problems are modeled. Recall that the matrices Δ^+ and Δ^- respectively describe unavoidable left and right leaf positions. Following Baatar *et al.*, we consider a third matrix W

describing *extra* leaf positions. Thus the leaf positions (unavoidable or extra) to be used in the decomposition can be read off from the matrices $\Delta^+ + W$ and $\Delta^- + W$. The matrices $\Delta^\pm + W$ play the role of the matrices $D^\pm$ used above for describing the standard decomposition algorithm. We require that $\Delta^\pm + W$ have constant row sums. This requirement is related to a previous remark: when the m-th row of a segment is zero then the positions ℓ_m and r_m of the leaves for that row should be specified. The constrained versions of BOT are modeled as integer programs whose variables are the entries of W and the total beam-on-time, which is simply the row sum of any row of $\Delta^+ + W$ or $\Delta^- + W$. It turns out that the resulting integer programs have particularly nice properties: their relaxations are integral and they can be solved by simple, direct algorithms. In fact, we show that solving these IPs amounts to finding a maximum value potential in an auxiliary network with integer arc lengths and no negative length cycle.

A network similar to one of the networks we consider was used by Kalinowski [10] to solve BOT under the interleaf motion constraint. A key result proved by Kalinowski (generalizing a result of Engel [7] for the unconstrained case) is that the optimum beam-on-time is the *maximum* length of a path in a network similar to our network D' (defined in Section 2.2). In contrast, we show that the optimum beam-on-time is the negative of the *minimum* length of a path in D'. Thus our approach unifies the approaches of both Baatar *et al.* and Kalinowski. In fact, our approach also considerably simplifies the previous approaches.

The rest of the paper is organised as follows: In Section 2 we describe how the constraints are modeled. In Section 2.1, we prove that BOT remains polynomial under the interleaf distance constraint. Section 2.2 focuses on the interleaf motion constraint. After recasting the results of Baatar *et al.* in our framework we indicate how to deal with both constraints at the same time. Finally, our fast algorithms for the constrained minimum beam-on-time problems are given in Section 3.

2 Modeling the Constrained Problems

2.1 The Interleaf Distance Constraint

In this section we consider the problem of minimizing the beam-on time with the interleaf distance constraint. This constraint asks that the distance between the end of two left (or two right) leaves cannot be bigger than a constant c; see Figure 5 for an example with $c = 2$. A segment $S = ([\ell_m, r_m])_{m=1,\ldots,M}$ respects the interleaf distance constraint if and only if the inequalities (3) and (4) hold.

Figure 5: When $c = 2$, the left segment respects the interleaf distance constraint while the right segment does not because the first left leaf is in position 4 and the last is in position 1.

In order to motivate our model, we first consider the particular case where the intensity matrix I

is such that all its rows have the same unavoidable beam-on time, that is, $T_1 = T_2 = \cdots = T_M =: T$ (cf. Equation (1)). For $k = 1, \ldots, T$, let ℓ_m^k (resp. r_m^k) denote the k-th unavoidable left (resp. right) leaf position in the m-th row of I.

Observation 2.1. *When all rows of the intensity matrix I have the same unavoidable beam-on time, say T, there exists a decomposition of I of beam-on-time T satisfying the interleaf distance constraint if and only if we have*

$$|\ell_m^k - \ell_{m'}^k| \leq c \qquad \textit{and} \qquad |r_m^k - r_{m'}^k| \leq c$$

for all $k \in \{1, \ldots, T\}$, and all m, $m' \in \{1, \ldots, M\}$. In particular, we can tell in polynomial time if such a decomposition of I exists.

Proof. To show the "if" direction we simply use the standard decomposition algorithm (cf. Algorithm 1) with $D^+ = \Delta^+$ and $D^- = \Delta^-$. Because the algorithm associates the first unmatched left leaf position with the first unmatched right leaf position for each row and we have $|\ell_m^k - \ell_{m'}^k| \leq c$ and $|r_m^k - r_{m'}^k| \leq c$ for all k, m and m', all the segments output by the algorithm respect the interleaf distance constraint.

We now prove the "only if" direction. Consider a decomposition of I satisfying the constraint with beam-on-time T. Then all unavoidable leaf positions and no further positions are used. Suppose, for instance, that there exist row indices m and m' and some integer k such that $\ell_m^k + c < \ell_{m'}^k$. Since we have $\ell_m^{k'} \leq \ell_m^k$ for $k' < k$ and $\ell_{m'}^{k''} \geq \ell_{m'}^k$ for $k'' > k$, all first k unavoidable left leaf positions in the m-th row are incompatible with all last $T - k + 1$ unavoidable left leaf positions in the m'-th row. Hence at most $k - 1$ left leaf positions in the m'-th row are compatible with the first k unavoidable left leaf positions in the m-th row, a contradiction. □

Let us consider now the general case. Consider a decomposition of I into segments. Just as we defined the difference matrix $\Delta = \Delta(I)$ of the intensity matrix I we can define a difference matrix $\Delta(S)$ for any segment S. This allows us to "differentiate" the considered decomposition of I in the following way:

$$I = \sum_{S \in \mathcal{S}} \alpha_S S \quad \Longrightarrow \quad \Delta(I) = \sum_{S \in \mathcal{S}} \alpha_S \Delta(S).$$

The last equation implies for the positive and negative parts of the considered matrices:

$$\sum_{S \in \mathcal{S}} \alpha_S \Delta^+(S) = \Delta^+(I) + X \quad \text{and} \quad \sum_{S \in \mathcal{S}} \alpha_S \Delta^-(S) = \Delta^-(I) + Y$$

for some non-negative integer matrices X and Y of size $M \times (N+1)$. Recall that $\Delta^+(I)$ and $\Delta^-(I)$ respectively describe the unavoidable left and right leaf positions. We interpret any entry x_{mn} of X as the number of *extra* left leaf positions equal to n in the m-th row, and similarly for the matrix Y. Now we have

$$0 = \sum_{S \in \mathcal{S}} \alpha_S \Delta(S) - \Delta(I) = \sum_{S \in \mathcal{S}} \alpha_S (\Delta^+(S) - \Delta^-(S)) - (\Delta^+(I) - \Delta^-(I)) = X - Y,$$

so the matrices X and Y are equal. That is, the number of extra leaf positions does not depend on the side (left or right). From now on, we will let $W = X = Y$. In conclusion, any decomposition of I determines a unique matrix W describing the extra leaf positions.

Conversely, if we have a non-negative integer matrix W of size $M \times (N+1)$ we can infer a decomposition of I as follows. We consider that the matrices $\Delta^+ + W$ and $\Delta^- + W$ respectively prescribe the left and right leaf positions of some decomposition. We then apply the standard decomposition algorithm with $D^+ = \Delta^+ + W$ and $D^- = \Delta^- + W$. This produces a decomposition of I because we have $D^+ - D^- = (\Delta^+ + W) - (\Delta^- + W) = \Delta$.

In conclusion, the matrix W implicitly defines a decomposition. As noted before, we have to require that $\Delta^+ + W$ (or $\Delta^- + W$) has constant row sums. This amounts to asking that the number of left (or right) leaf positions is the same for all rows. Therefore, all segments used in the decomposition have well defined left and right leaf positions for each row. Observe that the beam-on-time is then simply the row sum of any row of $\Delta^+ + W$ (or $\Delta^- + W$).

We can now state and prove a result generalizing Observation 2.1. As a direct consequence, we will obtain an integer programming formulation of BOT under the interleaf distance constraint.

Proposition 2.2. *Let Δ^+, Δ^- be defined as above. Consider a matrix W of extra leaf positions such that $\Delta^+ + W$ (and hence $\Delta^- + W$) has constant row sums. Then W induces a decomposition of I satisfying the interleaf distance constraint if and only if, for all $m, m' \in \{1, \ldots, M\}$, we have*

$$\sum_{n=1}^{j} (\delta^+_{mn} + w_{mn}) \leq \sum_{n=1}^{j+c} (\delta^+_{m'n} + w_{m'n}) \qquad \forall j \in \{1, \ldots, N-c+1\}; \tag{5}$$

$$\sum_{n=1}^{j} (\delta^-_{mn} + w_{mn}) \leq \sum_{n=1}^{j+c} (\delta^-_{m'n} + w_{m'n}) \qquad \forall j \in \{1, \ldots, N-c+1\}. \tag{6}$$

The beam-on-time of the decomposition induced by W is the row sum of any row of $\Delta^+ + W$.

Before proving the proposition, we offer an interpretation of Equations (5) and (6). Equation (5) says that the number of left leaf positions smaller than or equal to j in the m-th row does not exceed the number of left leaf positions smaller than or equal to $j + c$ in the m'-th row. The interpretation of (6) is similar.

Proof. Let T denote the row sum of any row of $\Delta^+ + W$, and let ℓ^k_m (resp. r^k_m) denote the k-th left (resp. right) leaf position for the m-th row, as described in $\Delta^+ + W$ (resp. $\Delta^- + W$). Then Equation (5) is equivalent to

$$|\ell^k_m - \ell^k_{m'}| \leq c \qquad \forall k \in \{1, \ldots, T\} \tag{7}$$

and Equation (6) is equivalent to

$$|r^k_m - r^k_{m'}| \leq c \qquad \forall k \in \{1, \ldots, T\} \tag{8}$$

Indeed, Equation (5) implies (7) because $\ell_m^k + c < \ell_{m'}^k$ for some k implies, for $j = \ell_m^k$,

$$\sum_{n=1}^{j}(\delta_{mn}^+ + w_{mn}) > \sum_{n=1}^{j+c}(\delta_{m'n}^+ + w_{m'n}).$$

Conversely, the latter inequality implies that there exists some k such that $\ell_m^k + c < \ell_{m'}^k$. A similar argument yields the equivalence of Equations (6) and (8). Now we can readily follow the proof of Observation 2.1 to conclude the proof of the result. □

Thanks to Proposition 2.2, we can model the beam-on-time problem under the interleaf distance constraint as follows:

$$\begin{array}{llll}
\text{(BOT-IDC)} & \min & T & \\
& \text{s.t.} & \sum_{n=1}^{N+1}(\delta_{mn}^+ + w_{mn}) = T & \forall m; \quad (9)\\
& & \sum_{n=1}^{j}(\delta_{mn}^+ + w_{mn}) \le \sum_{n=1}^{j+c}(\delta_{m'n}^+ + w_{m'n}) & \forall j, m, m' \ (m \neq m'); \quad (10)\\
& & \sum_{n=1}^{j}(\delta_{mn}^- + w_{mn}) \le \sum_{n=1}^{j+c}(\delta_{m'n}^- + w_{m'n}) & \forall j, m, m' \ (m \neq m'); \quad (11)\\
& & w_{mn} \ge 0 & \forall m, n;\\
& & w_{mn} \in \mathbb{Z} & \forall m, n. \quad (12)
\end{array}$$

In order to solve the above IP we first rewrite it by considering new variables π_0, π_{mj} for $1 \le m \le M$ and $1 \le j \le N+1$ and π_{N+2}. The relationship between the old and the new variables is as follows:

$$\begin{array}{rcll}
\pi_{mj} - \pi_0 & := & -\sum_{n=1}^{j} w_{mn} & \forall m \in \{1, \dots, M\},\ \forall j \in \{1, \dots, N+1\},\\
\pi_{N+2} - \pi_0 & := & -T. &
\end{array}$$

Furthermore, we replace the equality (9) by two inequalities. We thus obtain the following IP:

$$\begin{array}{llrcll}
\text{(BOT-IDC')} & \max & \pi_{N+2} - \pi_0 & & & \\
& \text{s.t.} & \pi_{N+2} - \pi_{m,N+1} & \le & -\sum_{n=1}^{N+1} \delta_{mn}^+ & \forall m;\\
& & \pi_{m,N+1} - \pi_{N+2} & \le & \sum_{n=1}^{N+1} \delta_{mn}^+ & \forall m;\\
& & \pi_{m',j+c} - \pi_{mj} & \le & \sum_{n=1}^{j+c} \delta_{m'n}^+ - \sum_{n=1}^{j} \delta_{mn}^+ & \forall j, m, m' \ (m \neq m'); \quad (13)\\
& & \pi_{m',j+c} - \pi_{mj} & \le & \sum_{n=1}^{j+c} \delta_{m'n}^- - \sum_{n=1}^{j} \delta_{mn}^- & \forall j, m, m' \ (m \neq m'); \quad (14)\\
& & \pi_{m1} - \pi_0 & \le & 0 & \forall m;\\
& & \pi_{mj} - \pi_{m,j-1} & \le & 0 & \forall m, j > 1;\\
& & \pi_{mj} & \in & \mathbb{Z} & \forall m, j. \quad (15)
\end{array}$$

Notice that the objective value of (BOT-IDC) is the negative of the objective value of (BOT-IDC').

The above formulation involves a system of difference constraints. It is well known that such a system can be represented by a network $D = (V, A)$ called *the constraint graph* (see Auhja, Magnanti and Orlin [2, p. 103]). The constraint graph has one node i for each variable x_i and one arc (i, j) of length b for each constraint of the type $x_j - x_i \leq b$. In our case, the set of nodes is

$$V := \{0\} \cup \{(m, j) : 1 \leq m \leq M,\ 1 \leq j \leq N + 1\} \cup \{N + 2\}.$$

We denote the set of arcs by A and the length of each arc $a \in A$ by $\ell(a)$. To avoid parallel arcs, we create only one arc for each pair of constraints (13) and (14) with the same parameters; the length of this arc is the minimum of the two upper bounds.

The new variables determine a potential in the network $D = (V, A)$ with respect to the length function $\ell : A \to \mathbb{Z}$. Thus solving BOT under the interleaf distance constraint amounts to finding a potential π whose value $\pi_{N+2} - \pi_0$ is maximum. The optimal beam-on-time is simply the negative of the distance (that is, the length of a shortest path) from vertex 0 to vertex $N+2$ in the network.

The following remarks are in order. Note that the only (simple) cycles in the network D contain two opposite arcs and have length zero. So there are no negative length cycles in the network. In fact D is essentially acyclic. Because all lengths are integral the integrality constraint (15) can be removed from problem (BOT-IDC'). Indeed, the constraint matrix of (BOT-IDC') is totally unimodular since each row of the constraint matrix has exactly one 1, one -1 and 0 everywhere else, and all right hand sides are integral. It follows that the integrality constraint (12) can also be removed from problem (BOT-IDC).

Because an optimal potential can be computed in polynomial time, for example, via the Bellman-Ford method [4, 8], we derive the following result which is our main result. For faster algorithms, see Section 3.

Theorem 2.3. *Solving BOT under the interleaf distance constraint can be done in polynomial time.* □

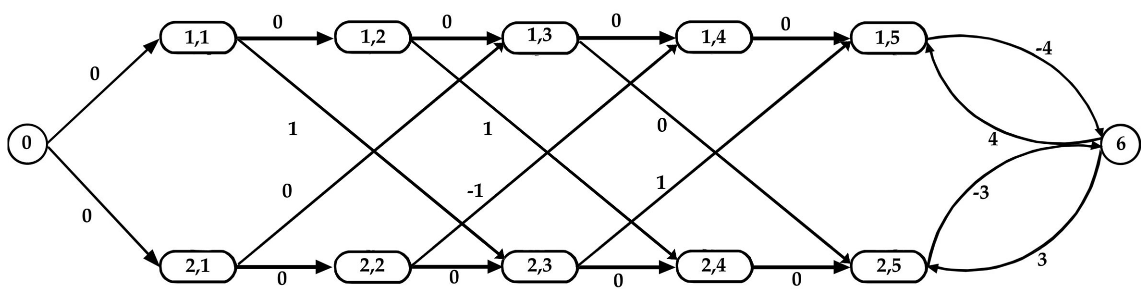

Figure 6: Example of network D for the interleaf distance constraint.

To conclude this section, we briefly discuss an example for a constant equal to 2. Let I be the following intensity matrix:

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

We want to find an optimal decomposition of I which respects the interleaf distance constraint for a constant $c = 2$. The network corresponding to I is given in Figure 6. After computing an optimal potential we obtain

$$W = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix},$$

which yields the following optimal decomposition:

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

2.2 The Interleaf Motion Constraint

In this section we consider the interleaf motion constraint. Recall that this constraint forbids the left leaf in some row to overlap the right leaf in an adjacent row (see Figure 7).

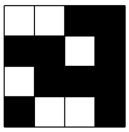

Figure 7: A segment that does not satisfy the interleaf motion constraint because the left leaf in the second row overlaps the right leaf in the third row.

We can use exactly the same ideas as for the interleaf distance constraint for setting up a model in this case. Baatar, Hamacher, Ehrgott and Woeginger [3] proved the following result.

Proposition 2.4. *Consider a matrix W of extra leaf positions such that $\Delta^+ + W$ (and hence $\Delta^- + W$) has constant row sums. Then W induces a decomposition of I satisfying the interleaf motion constraint if and only if, for all $m \in \{2, \dots, M\}$ and all $j \in \{1, \dots, N+1\}$, we have*

$$\sum_{n=1}^{j} (\delta^+_{m-1,n} + w_{m-1,n}) \geq \sum_{n=1}^{j} (\delta^-_{mn} + w_{mn}), \tag{16}$$

$$\sum_{n=1}^{j} (\delta^+_{mn} + w_{mn}) \geq \sum_{n=1}^{j} (\delta^-_{m-1,n} + w_{m-1,n}). \tag{17}$$

The beam-on-time of the decomposition induced by W is the row sum of any row of $\Delta^+ + W$.

To obtain an IP model similar to (BOT-IDC) for BOT under the interleaf motion constraint it suffices to replace the two constraints (10) and (11) by the constraints (16) and (17). We obtain an IP formulation which we denote by (BOT-IMC). A simpler IP formulation (BOT-IMC') can be obtained by the same change of variables as above. Alternatively, it suffices to replace the two constraints (13) and (14) in (BOT-IDC') by the following two constraints:

$$\pi_{m-1,j} - \pi_{mj} \leq \sum_{n=1}^{j} \delta^+_{m-1,n} - \sum_{n=1}^{j} \delta^-_{mn}, \ \forall j, \forall m \geq 2 \tag{18}$$

$$\pi_{mj} - \pi_{m-1,j} \leq \sum_{n=1}^{j} \delta^+_{mn} - \sum_{n=1}^{j} \delta^-_{m-1,n}, \ \forall j, \forall m \geq 2. \tag{19}$$

We obtain an IP formulation involving a system of difference constraints. We denote the resulting formulation (BOT-IMC'). Once again the formulation models a maximum value potential problem in some network $D' = (V', A')$.

As discussed in the introduction, a network similar to D' appears in Kalinowski [10]. The new network D' has cycles involving two, three or more arcs but any of these has non-negative length (recall that cycles have no repeated nodes). Indeed, for the cycles containing two arcs of the form $((m, j), (m-1, j))$ and $((m-1, j), (m, j))$, we have

$$\begin{aligned} \ell((m,j),(m-1,j)) + \ell((m-1,j),(m,j)) &= \sum_{n=1}^{j} \delta^+_{m-1,n} - \sum_{n=1}^{j} \delta^-_{m-1,n} + \sum_{n=1}^{j} \delta^+_{mn} - \sum_{n=1}^{j} \delta^-_{mn} \\ &= i_{m-1,j} + i_{mj} \geq 0. \end{aligned}$$

An easy computation shows that all other cycles have length zero. It follows that BOT under the interleaf motion constraint can again be solved in polynomial time, for example, by the Bellman-Ford method [4, 8]. This yields a new and simpler derivation of a main result of Baatar *et al.* [3] (see also Kalinowski [10]).

Theorem 2.5 (Baatar *et al.* [3])**.** *Solving BOT under the interleaf motion constraint can be done in polynomial time.*

Moreover, the IP formulation (BOT-IMC') and thus (BOT-IMC) both have integral relaxations.

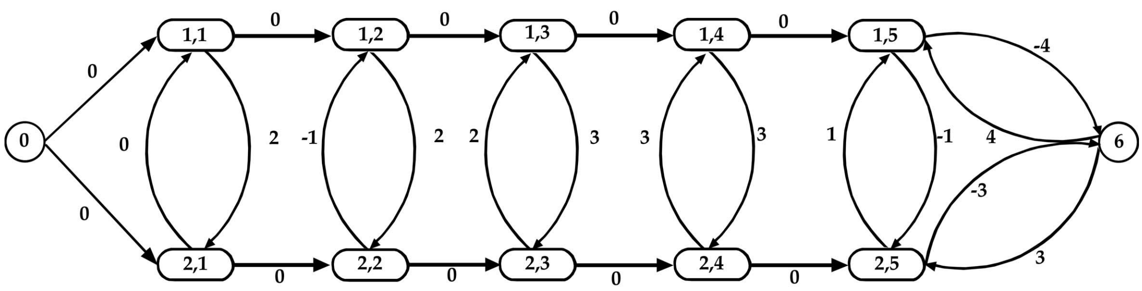

Figure 8: Example of network D' for the interleaf motion constraint.

We now continue the example considered in the previous section and give the corresponding network D' in Figure 8. This time after computing a maximum value potential, we find

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

To conclude this section we explain how to combine both constraints. In this case we have to add the new constraints (18) and (19) to the model (BOT-IDC'), while keeping the constraints (13) and (14). This corresponds to adding new arcs in the network. Once again the resulting network, which we denote by $D'' = (V'', A'')$, has no negative cycles and the IPs have integral relaxations.

Theorem 2.6. *Solving BOT under the interleaf distance and interleaf motion constraints can be done in polynomial time.*

3 Fast Algorithms

3.1 The Interleaf Distance Constraint

In Section 2.1 we showed that solving BOT under the interleaf distance constraint amounts to finding a potential π in the network D whose value $\pi_{N+2} - \pi_0$ is maximum. The optimal beam-on-time is exactly the negative of the length of a shortest path from vertex 0 to vertex $N+2$ in the network. As the network D is essentially acyclic we can adapt the standard dynamic programming algorithm due to Morávek [14] to compute an optimal potential in time $O(|A|) = O(M^2N)$. However, as we show here, an optimal potential can be computed in time $O(MN)$. When we take into account the time it takes to compute W and produce the decomposition from the potential via the standard decomposition algorithm, we obtain an algorithm for BOT under the interleaf distance constraint whose running time is $O(MN + KM)$.

Our algorithm for computing an optimal potential first initializes to 0 π_0 and π_{mj} for $m = 1, \ldots, M$ and $j = 1, \ldots, c$. Next we consider the vertices (m, j) for $j = c+1$, then for $j = c+2$, and so on until $j = N$. Each time a new vertex (m, j) is considered we compute its potential as follows:

$$
\begin{aligned}
\pi_{mj} &= \min \left\{ \pi_{m,j-1} + \ell\left(\pi_{m,j-1}, \pi_{mj}\right), \quad \min \left\{ \pi_{m',j-c} + \ell\left(\pi_{m',j-c}, \pi_{mj}\right) : m' \neq m \right\} \right\} \\
&= \min \left\{ \pi_{m,j-1}, \ \min \left\{ \pi_{m',j-c} + \sum_{n=1}^{j} \delta^+_{mn} - \sum_{n=1}^{j-c} \delta^+_{m'n}, \right.\right. \\
&\qquad\qquad \left.\left. \pi_{m',j-c} + \sum_{n=1}^{j} \delta^-_{mn} - \sum_{n=1}^{j-c} \delta^-_{m'n} : m' \neq m \right\} \right\}. \qquad (20)
\end{aligned}
$$

For the $M+1$ remaining vertices $(1, N+1)$, $\ldots$, $(M, N+1)$ and $N+2$ we compute all the potentials at the same time.

Let $\Sigma^+ = \left(\sigma^+_{mj}\right)$ and $\Sigma^- = \left(\sigma^-_{mj}\right)$ be the $M \times (N+1)$ matrices defined as:

$$
\sigma^+_{mj} = \sum_{n=1}^{j} \delta^+_{mn} \quad \text{and} \quad \sigma^-_{mj} = \sum_{n=1}^{j} \delta^-_{mn}.
$$

Instead of computing the inner minimum each time in Equation (20), which would lead to a $O(M^2N)$ algorithm for finding the potential, we compute only once the minimum μ^+_{j-c} of the expression $\pi_{m',j-c} - \sigma^+_{m',j-c}$ for $m' = 1, \ldots, M$ and the minimum μ^-_{j-c} of the expression $\pi_{m',j-c} -$

$\sigma^-_{m',j-c}$ for $m' = 1, \dots, M$. Therefore, we have

$$\pi_{mj} \;=\; \min\ \left\{\ \pi_{m,j-1},\ \mu^+_{j-c} + \sigma^+_{mj},\ \mu^-_{j-c} + \sigma^-_{mj}\ \right\}.$$

The resulting algorithm is formally described below (Algorithm 2).

Algorithm 2 Algorithm for finding an optimal potential (interleaf distance constraint).

Input: The intensity matrix I.
Output: An optimal potential π in network D.

```
Compute the matrices Δ⁺, Δ⁻, Σ⁺ and Σ⁻ from I
π_0 ← 0
for all m ∈ {1,...,M} and j ∈ {1,...,c} do
  π_mj ← 0
end for
for all j ∈ {c+1,...,N} do
  μ⁺_{j-c} ← min{π_{m',j-c} − σ⁺_{m',j-c} : m' = 1,...,M}
  μ⁻_{j-c} ← min{π_{m',j-c} − σ⁻_{m',j-c} : m' = 1,...,M}
  for all m ∈ {1,...,M} do
    π_mj ← min{π_{m,j-1}, μ⁺_{j-c} + σ⁺_mj, μ⁻_{j-c} + σ⁻_mj}
  end for
end for
μ⁺_{N+1-c} ← min{π_{m',N+1-c} − σ⁺_{m',N+1-c} : m' = 1,...,M}
μ⁻_{N+1-c} ← min{π_{m',N+1-c} − σ⁻_{m',N+1-c} : m' = 1,...,M}
π_{N+2} ← min{−σ⁺_{m,N+1} + min{π_{m,N}, μ⁺_{N+1-c} + σ⁺_{m,N+1}, μ⁻_{N+1-c} + σ⁻_{m,N+1}} : m = 1,...,M}
for all m ∈ {1,...,M} do
  π_{m,N+1} ← π_{N+2} + σ⁺_{m,N+1}
end for
```

Algorithm 2 finds an optimal potential in time $O(MN)$. To find the decomposition we first construct the matrix W using $w_{mj} = \pi_{m,j-1} - \pi_{mj}$ (for $j = 1$ we let $\pi_{m,j-1} = \pi_0 = 0$). This can also be done in $O(MN)$ time. Then to produce the decomposition we use Algorithm 1 with $D^+ = \Delta^+ + W$ and $D^- = \Delta^- + W$. Because the complexity of the standard decomposition algorithm is $O(KM)$, we obtain the following result. Recall that K denotes the number of matrices output by the decomposition algorithm.

Proposition 3.1. *The minimum beam-on-time problem under the interleaf distance constraint can be solved in time $O(MN + KM)$.*

3.2 The Interleaf Motion Constraint

We proved in Section 2.2 that BOT under the interleaf motion constraint can be solved by determining an optimal potential in the network D'. Because the network is highly structured we can compute an optimal potential in time $O(MN \log M)$. Hence, if we add the time it takes to compute W from the potential and produce the decomposition via the standard decomposition algorithm, we conclude that BOT under the interleaf motion constraint can be solved in time $O(MN \log M + KM)$. In contrast, the algorithm proposed by Baatar *et al.* has complexity $O(M^2N + KM)$.

The basic principle of our algorithm is the same as for Algorithm 2, namely, we consider the vertices (m, j) by nondecreasing value of j. The main difference is that in the network D' all the vertices (m, j) with a fixed j are in the same strongly connected component. So the potentials of these vertices should be determined simultaneously. This is done by solving the following subproblem.

Let $a_2, \ldots, a_M$ and $b_2, \ldots, b_M$ be $2(M-1)$ integers such that $a_m \leq b_m$ for all $m \in \{2, \ldots, M\}$. We seek a vector $x \in \mathbb{Z}^M$ satisfying

$$a_m \leq x_m - x_{m-1} \leq b_m \qquad \forall m \in \{2, \ldots, M\}, \tag{21}$$

$$x_m \leq 0 \qquad \forall m \in \{1, \ldots, M\} \tag{22}$$

that simultaneously maximizes all its coordinates. Note that such a vector exists and is unique. As we explain below, the subproblem can be solved in time $O(M \log M)$.

First we initialize x_1 to 0 and x_m to its upper bound $x_{m-1} + b_m$ for $m = 2, \ldots, M$. If the resulting vector x is non-positive then it is the optimal solution and we are done. Otherwise we compute any index p such that $x_p = \max\{x_m : m = 1, \ldots, M\}$ and we decrease all coordinates of x by x_p. So now x is non-positive and its p-th coordinate equals 0. Next we try to increase the other coordinates of x. Because the coordinates of x whose index is greater than or equal to p cannot be increased, we consider these values to be fixed. When we simultaneously increase all variables x_m with $m < p$, one of the two following events eventually occurs. First, some coordinate x_q of x can become 0. Second, x_{p-1} can be increased up to the point where we have $a_p = x_p - x_{p-1}$. Any of these two events blocks us from increasing the variables x_m with $m < p$. If the first event occurs first, we replace p by q and continue increasing the values x_m with $m < p$. Otherwise, the second event occurs first, and we replace p by $p - 1$ and continue increasing the values x_m with $m < p$. We stop as soon as p equals 1. The final x vector is the optimal solution. This is because each coordinate of x is either zero or maximal with respect to the value of one of the neighboring coordinates of x.

The algorithm described above can be implemented to run in time $O(M \log M)$; see Algorithm 3. Algorithm 4 uses Algorithm 3 to find an optimal potential in $D' = (V', A')$ in time $O(MN \log M)$. The next result follows.

Proposition 3.2. *The minimum beam-on-time problem under the interleaf motion constraint can be solved in time* $O(MN \log M + KM)$.

A straightforward modification of Algorithm 4 yields an algorithm for computing an optimal potential in the network D''. (It suffices to change the initializations of the upper bounds o_m in the obvious way.)

Proposition 3.3. *The minimum beam-on-time problem under the interleaf distance and interleaf motion constraints can be solved in time* $O(MN \log M + KM)$.

Algorithm 3 Algorithm for solving the subproblem.

Input: integers $a_2, \ldots, a_M$ and $b_2, \ldots, b_M$ such that $a_m \leq b_m$ for all $m \in \{2, \ldots, M\}$.

Output: an integer vector x with m coordinates satisfying Equations (21) and (22), and maximizing all its coordinates simultaneously.

```
h ← 0
L ← ∅ {L is an associative array with its keys and values integer}
Add (1, h) to L {The first coordinate of an element of L is its key and the second its value}
for all m ∈ {2, ..., M} do
  h ← h + b_m
  Add (m, h) to L
end for
Sort the elements of L in such a way that their second coordinates are non-increasing
(p, δ) ← any element of L with maximum value
for all m ∈ {p, ..., M} do
  x_m ← L[m] − δ
  Remove the element with key m from L
end for
while p > 1 do
  (q, h) ← any element of L with maximum value
  δ_1 ← 0 − (h − δ) {Maximum increase for x_q}
  δ_2 ← (x_p − a_p) − (L[p − 1] − δ) {Maximum increase for x_{p−1}}
  if δ_1 ≤ δ_2 then
    δ ← δ − δ_1
    for all m ∈ {q, ..., p − 1} do
      x_m ← L[m] − δ
      Remove the element with key m from L
    end for
    p ← q
  else
    δ ← δ − δ_2
    x_{p−1} ← L[p − 1] − δ
    Remove the element with key p − 1 from L
    p ← p − 1
  end if
end while
```

4 Acknowledgments

We thank Martine Labbé for suggesting us to work on the beam-on-time problem in the early stage of this research project. We thank Pierre Scalliet and Stefaan Vynckier from the radiotherapy service of the “Cliniques universitaires Saint-Luc” (Brussels) and Stéphane Simon from the “Institut Bordet” (Brussels) for several discussions. We thank Horst Hamacher and Matthias Ehrgott for their interest in our results. We also thank Arvind Kumar for sending us his joint paper with Ravindra K. Ahuja and Srijit Kamath [13]. Finally, we want to thank the two anonymous referees for their constructive remarks.

Algorithm 4 Algorithm for finding an optimal potential (interleaf motion constraint).

Input: The intensity matrix I.
Output: An optimal potential π.

Compute the matrices Δ^+, Δ^-, Σ^+ and Σ^- from I
$\pi_0 \leftarrow 0$
for all $m \in \{1, \dots, M\}$ **do**
  $\pi_{m1} \leftarrow 0$
end for
for all $j \in \{2, \dots, N\}$ **do**
  for all $m \in \{1, \dots, M\}$ **do**
    $o_m \leftarrow \pi_{m,j-1}$ {Upper bound on π_{mj} from the vertices (m', j') with $j' < j$}
  end for
  for all $m \in \{2, \dots, M\}$ **do**
    $a_m \leftarrow \sigma^-_{mj} - \sigma^+_{m-1,j} + o_m - o_{m-1}$
    $b_m \leftarrow \sigma^+_{mj} - \sigma^-_{m-1,j} + o_m - o_{m-1}$
  end for
  Call Algorithm 3 to solve the subproblem defined by $a_2, \dots, a_M$ and $b_2, \dots, b_M$
  for all $m \in \{1, \dots, M\}$ **do**
    $\pi_{m,j} \leftarrow x_m + o_m$
  end for
end for
for all $m \in \{1, \dots, M\}$ **do**
  $o_m \leftarrow \pi_{m,N}$ {Upper bound on $\pi_{m,N+1}$ from the vertices (m', j') with $j' < N + 1$}
end for
$\pi_{N+2} \leftarrow \min\{-\sigma^+_{m,N+1} + o_m : m = 1, \dots, M\}$
for all $m \in \{1, \dots, M\}$ **do**
  $\pi_{m,N+1} \leftarrow \pi_{N+2} + \sigma^+_{m,N+1}$
end for

References

[1] R.K. Ahuja and H.W. Hamacher, A network flow algorithm to minimize beam-on time for unconstrained multileaf collimator problems in cancer radiation therapy, Networks 45 (2005), 36–41.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, Networks flows, Prentice Hall, Englewood Cliffs, NJ, 1993.

[3] D. Baatar, H.W. Hamacher, M. Ehrgott, and G.J. Woeginger, Decomposition of integer matrices and multileaf collimator sequencing, Discrete Appl Math 152 (2005), 6–34.

[4] R. Bellman, On a routing problem, Quart Appl Mat 16 (1958), 87–90.

[5] N. Boland, H.W. Hamacher, and F. Lenzen, Minimizing beam-on time in cancer radiation treatment using multileaf collimators, Networks 43 (2004), 226–240.

[6] T.R. Bortfeld, D.L. Kahler, T.J. Waldron, and A.L. Boyer, X-ray field compensation with multileaf collimators, Int J Radiat Oncol Biol Phys 28 (1994), 723–730.

[7] K. Engel, A new algorithm for optimal multileaf collimator field segmentation, Discrete Appl Math 152 (2005), 35–51.

[8] L.R. Ford, Jr, Network flow theory, The RAND Corporation, Paper P-923, Santa Monica, CA, 1956.

[9] H.W. Hamacher and K.-H. Küfer, Inverse radiation therapy planning – A multiple objective optimization approach, Discrete Appl Math 118 (2002) 145–161.

[10] T. Kalinowski, A duality based algorithm for multileaf collimator field segmentation with interleaf collision constraint, Discrete Appl Math 152 (2005), 52–88.

[11] T. Kalinowski, "Multileaf collimator shape matrix decomposition," Optimization in Medicine and Biology, G.J. Lim (Editor), Auerbach Publishers Inc., 2008, pp. 249–282.

[12] S. Kamath, S. Sahni, J. Li, J. Palta, and S. Ranka, Leaf sequencing algorithms for segmented multileaf collimation, Phys Med Biol 48 (2003), 307–324.

[13] A. Kumar, R.K. Ahuja, and S. Kamath, Network-based models for segmented multileaf collimator sequencing, Working paper, 2007.

[14] J. Morávek, A note upon minimal path problem, J Math Anal Appl 30 (1970), 702–717.

[15] W. Schlegel and A. Mahr, 3D conformal radiation therapy: Multimedia introduction to methods and techniques, Springer Verlag, Heidelberg, 2002.

[16] R.A.C. Siochi, Minimizing static intensity modulation delivery time using an intensity solid paradigm, Int J Radiat Oncol Biol Phys 43 (1999), 671–680.

[17] P. Xia and L.J. Verhey, Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple segments, Med Phys 25 (1998), 1424–1434.