Minimum Cardinality Matrix Decomposition into Consecutive-Ones Matrices: CP and IP Approaches

Davaatseren Baatar[1], Natashia Boland[1], Sebastian Brand[2], and Peter J. Stuckey[2]

[1] Department of Mathematics, University of Melbourne, Australia
[2] NICTA Victoria Research Lab, Department of Comp. Sci. and Soft. Eng.
University of Melbourne, Australia

Abstract. We consider the problem of decomposing an integer matrix into a positively weighted sum of binary matrices that have the consecutive-ones property. This problem is well-known and of practical relevance. It has an important application in cancer radiation therapy treatment planning: the sequencing of multileaf collimators to deliver a given radiation intensity matrix, representing (a component of) the treatment plan.

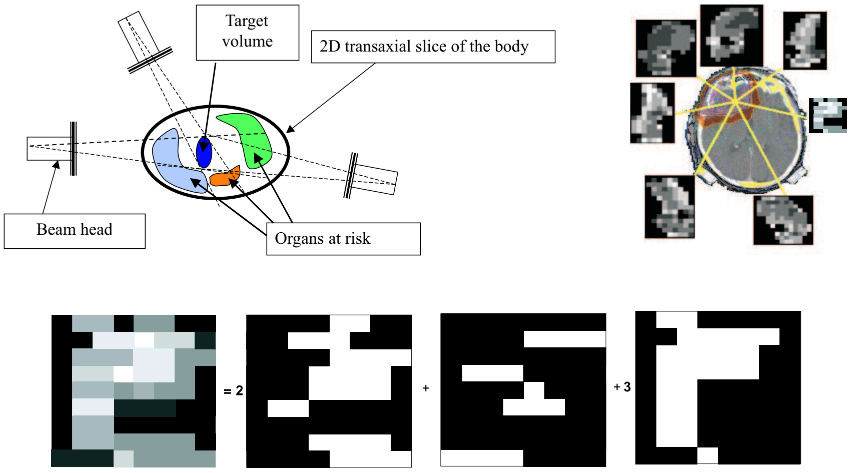
Two criteria characterise the efficacy of a decomposition: the *beam-on time* (length of time the radiation source is switched on during the treatment), and the *cardinality* (the number of machine set-ups required to deliver the planned treatment).

Minimising the former is known to be easy. However finding a decomposition of minimal cardinality is NP-hard. Progress so far has largely been restricted to heuristic algorithms, mostly using linear programming, integer programming and combinatorial enumerative methods as the solving technologies. We present a novel model, with corresponding constraint programming and integer programming formulations. We compare these computationally with previous formulations, and we show that constraint programming performs very well by comparison.

1 Introduction

The problem of decomposing an integer matrix into a weighted sum of binary matrices has received much attention in recent years, largely due to its application in radiation treatment for cancer.

Intensity-modulated radiation therapy (IMRT) has been increasingly used for the treatment of a variety of cancers [17]. This treatment approach employs two devices that allow higher doses of radiation to be administered to the tumour, while decreasing the exposure of sensitive organs (Fig. 1). The first is that the source of radiation can be rotated about the body of the patient: by positioning the tumour at a “focal point”, and aiming the radiation beam at this point from various angles, the tumour receives a high dose from all angles, while the surrounding tissue only gets high exposure from some angles.



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

Fig. 1. Intensity-modulated radiotherapy

The second is more subtle, and involves repeated exposures from the same angle, where the uniform-intensity rectangular field of radiation delivered by the radiation source is "shaped" in a different way for each exposure, and each exposure can be for a different length of time. This process builds up a complex profile of received radiation in the patient's body, effectively converting the uniform radiation field delivered by the machine to an intensity-modulated field. The latter is usually described by discretising the 2-dimensional rectangular field, and specifying a radiation intensity level in each discrete element, representing the total length of time for which that element should be exposed to radiation.

A *treatment plan* for a single IMRT treatment session with a patient thus typically consists of a set of angles, together with a matrix for each angle, known as the *intensity matrix*, which represents the modulated field to be delivered at that angle. Typically the intensity is scaled so that the entries in the intensity matrix are integer. Indeed, they are usually quite small integers. Finding a good treatment plan is a challenging problem in its own right, and has been the subject of a great deal of research. We recommend the reader refer to the papers [13, 9, 15] and references therein.

In this paper, we assume a treatment plan is given, and focus on the delivery of the modulated field (intensity matrix) at a given angle. IMRT can be delivered by a variety of technologies: here we focus on its delivery via a machine known as a multileaf collimator, operating in “step-and-shoot” mode [7]. This machine delivers a rectangular field of radiation, of uniform intensity, that can be shaped through partial occlusion of the field by lead rods, or “leaves”. These are positioned horizontally on the left and right side of the field, and can slide laterally across the field to block the radiation, and so shape the field. The discretisation giving rise to the intensity matrix is taken to be compatible with the leaf widths. In step-and-shoot mode, the leaves are moved into a specified position, the radiation source switched on for a specified length of time and then switched off, the leaves moved to a new position, and so on (Fig. 1).

The shaped radiation field delivered by the leaves in each position can be represented as a binary matrix, with 1's in elements exposed in that position, and 0's in elements covered by the leaves (Fig. 1). The structure of the machinery ensures that all 1's in any row occur in a consecutive sequence: the matrix has the consecutive-ones property. The length of time radiation is applied to the shaped field is called its beam-on time. To correctly deliver the required intensity matrix, the matrices corresponding to the shaped fields, weighted with their beam-on times, must sum to the intensity matrix.

This motivates the following problem specification.

2 Problem Specification and Related Work

Let I be an $m \times n$ matrix of non-negative integers (the intensity matrix). The problem is to find a decomposition of I into a positive linear combination of binary matrices that have the consecutive-ones property. Often the radiation delivery technology imposes other constraints on the matrices, but here we focus on the simplest form, in which only the consecutive-ones property is required. For convenience, we use the abbreviation C1 for a binary consecutive-ones matrix. We also refer to a shaped field, represented by a C1 matrix, as a *pattern*.

Formally, we seek positive integer coefficients b_k (the beam-on times) and C1 matrices X_k (the patterns), such that

$$I = \sum_{k \in \Omega} b_k X_k \tag{1}$$

where Ω is the index set of the binary matrices X_k, and for $k \in \Omega$:

$$X_{k,i,j_L} = 1 \wedge X_{k,i,j_R} = 1 \;\rightarrow\; X_{k,i,j_M} = 1 \tag{2}$$

for all $1 \leqslant j_L < j_M < j_R \leqslant n$ and all $i = 1, \ldots, m$.

Example 1. Consider the matrix

$$I = \begin{pmatrix} 2 & 5 & 3 \\ 3 & 5 & 2 \end{pmatrix}.$$

Two decompositions are

$$\begin{aligned} D_1 &= 1\begin{pmatrix}0&1&1\\1&1&0\end{pmatrix} + 2\begin{pmatrix}1&1&1\\1&1&1\end{pmatrix} + 2\begin{pmatrix}0&1&0\\0&1&0\end{pmatrix} \quad \text{and} \\ D_2 &= 2\begin{pmatrix}1&1&0\\0&1&1\end{pmatrix} + 3\begin{pmatrix}0&1&1\\1&1&0\end{pmatrix}, \end{aligned}$$

that is, we have $I = D_1 = D_2$. ◊

We denote by B and K the sum of coefficients b_k and the number of matrices X_k used in the decomposition (1), respectively, i.e.

$$B = \sum_{k \in \Omega} b_k \qquad \text{and} \qquad K = |\{\, b_k \mid \ k \in \Omega \,\}|\,.$$

We call B the *total beam-on time* and K the *cardinality* of the decomposition. The efficacy of a decomposition is characterised by its values for B and K: the smaller these values the better. In Example 1, the decompositions have the values $B_1 = B_2 = 5$, $K_1 = 3$ and $K_2 = 2$; so D_2 is preferred.

The problem of finding a decomposition that minimises B can be solved in polynomial time using linear programming or combinatorial algorithms [1, 8, 10, 3]. However, it is possible for a decomposition to have minimal B but large K; indeed algorithms for minimising B tend to produce solutions with much larger K values than is necessary.

In radiation therapy, clinical practitioners would prefer solutions that minimise B, while ensuring K is as small as possible, i.e. they would prefer a lexicographically minimum solution, minimising B first and then K, written as $\mathit{lex\_min}(B, K)$. Since minimising B is easy, its minimal value, which we denote by B^*, is readily computable. The problem then becomes one of minimising K subject to the constraint that $B = B^*$. Although this problem, too, is NP-hard (it follows directly from the proof of NP-hardness of the problem of minimising K alone, given in [3]), it is hoped that solution methods effective in practice can be developed.

In the last decade, dozens of heuristic algorithms have indeed been developed, for example [1, 8, 10, 6, 3]; approximation algorithms are studied in [4]. Some of these attempt to find solutions in which both B and K are "small", while some seek low cardinality solutions while ensuring $B = B^*$ is fixed. An exact algorithm for the $\mathit{lex\_min}(B, K)$ problem has also been developed [10]: it is a highly complex, specialised enumerative algorithm that appears to carry out similar steps to those that might be expected in a constraint programming approach.

However the development of tractable exact formulations has lagged behind. Several exact integer programming models were introduced in [2] and [11] in order to solve the $\mathit{lex\_min}(B, K)$ problem, but these were either not tested computationally or were able to solve only small problems in reasonable CPU time. In this paper we develop a new model, that we refer to as the *Counter Model*. We derive both an integer programming formulation and a constraint programming method, and test both of these computationally against previous integer programming models. Our integer programming formulation performs substantially better than existing formulations, and the constraint programming approach provides the best computational results overall.

In the remainder of this paper, we first briefly review the existing integer programming formulations, and then present the Counter Model, our new integer programming formulation, and our constraint programming method. We then provide a computational comparison of these, and make our conclusions.

3 Existing Integer Programming Formulations

The central issue in modelling the $\mathit{lex\_min}(B, K)$ problem is that K, the cardinality of the decomposition, is unknown, and yet the natural variable indices depend on it. The two integer linear programming models in the current literature that can be used for the $\mathit{lex\_min}(B, K)$ problem take different approaches in tackling this issue. [11] overcome it by indexing according to radiation units; [2] instead calculates an upper bound on K. Here we give descriptions of these models and some additional symmetry breaking constraints.

Notation. The range expression $[a..b]$ with integers a, b denotes the integer set $\{\, e \mid a \leqslant e \leqslant b \,\}$.

3.1 The Unit Radiation Model

The model of [11] focuses on individual units of radiation. It is based on the assumption that the total beam-on time is fixed, in our case to B^*. What is not known is: for each of the B^* units of radiation, what pattern should be used for the delivery of that unit? In the model, binary variables $d_{t,i,j}$ are used to indicate whether the element (i, j) is exposed in the tth pattern corresponding to the tth unit of radiation, for $t \in [1..B^*]$. They are linked to the intensity matrix by

$$I_{ij} = \sum_{t=1}^{B^*} d_{t,i,j}, \qquad \text{for all } i \in [1..m], j \in [1..n]. \tag{3}$$

The leaf structure in the pattern is captured by binary variables:

$$p_{t,i,j} = \begin{cases} 1 & \text{if the right leaf in row } i \text{ of pattern } t \text{ covers column } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$\ell_{t,i,j} = \begin{cases} 1 & \text{if the left leaf in row } i \text{ of pattern } t \text{ covers column } j, \\ 0 & \text{otherwise,} \end{cases}$$

for all $t \in [1..B^*]$, $i \in [1..m]$, $j \in [1..n]$. The relationship between these three sets of binary variables is given by

$$p_{t,i,j} + \ell_{t,i,j} = 1 - d_{t,i,j} \qquad \text{for all } t \in [1..B^*], i \in [1..m], j \in [1..n], \tag{4}$$

and

$$\begin{aligned} p_{t,i,j} &\leqslant p_{t,i,j+1}, \\ \ell_{t,i,j+1} &\leqslant \ell_{t,i,j} \end{aligned} \qquad \text{for all } t \in [1..B^*], i \in [1..m], j \in [1..n-1]. \tag{5}$$

These constraints ensure that the d_t induce a C1 matrix.

Under these constraints, the indices t can be permuted to create equivalent solutions. Thus, the model is "free" to order the patterns so that identical patterns appear consecutively. To minimise the number of different patterns in a solution, the number of times adjacent patterns are different can be minimised. That patterns t and $t+1$ differ is reflected in the binary variable g_t, and the sum of these variables corresponds to the number of patterns by

$$K = 1 + \sum_{t=1}^{B^*-1} g_t. \tag{6}$$

Minimising this sum ensures that identical patterns appear consecutively. Each unique pattern yields a C1 matrix for the decomposition; the associated beam-on time is given by the number of copies of this pattern among the d_t. [11] tally values for g using binary additional variables:

$$\begin{aligned} c_{t,i,j} &= \begin{cases} 1 & \text{if } d_{t,i,j} = 1 \text{ and } d_{t+1,i,j} = 0, \\ 0 & \text{otherwise,} \end{cases} \\ u_{t,i,j} &= \begin{cases} 1 & \text{if } d_{t,i,j} = 0 \text{ and } d_{t+1,i,j} = 1, \\ 0 & \text{otherwise,} \end{cases} \\ s_{t,i,j} &= \begin{cases} 1 & \text{if } d_{t,i,j} \neq d_{t+1,i,j}, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

for all $t \in [1 \,..\, B^* - 1]$, $i \in [1..m]$, $j \in [1..n]$. The relationship of these variables is established by the linear constraints

$$\begin{aligned} & - c_{t,i,j} \leqslant d_{t+1,i,j} - d_{t,i,j} \leqslant u_{t,i,j}, \\ & u_{t,i,j} + c_{t,i,j} = s_{t,i,j}, \\ & \sum_{i=1}^{m} \sum_{j=1}^{n} s_{t,i,j} \leqslant mng_t, \\ & \text{for all } t \in [1 \,..\, B^* - 1], i \in [1..m], j \in [1..n]. \end{aligned} \tag{7}$$

The original model in [11] does not contain symmetry-breaking constraints. We added symmetry breaking constraints as follows. We wish to enforce that the matrices appear in order of non-increasing beam-on time. This means the pattern groups should appear in order of non-increasing size, which is to say that no (possibly empty) sequence of 0's enclosed by 1's and followed by a longer sequence of 0's occurs in the g vector extended by $g_0 = 1$. This can be enforced by

$$\sum_{v=r}^{r+t} g_v - 1 \leqslant \sum_{v=r+t+1}^{r+2t} g_v \quad \text{for all } r \in [0 \mathinner{..} B^* - 3], t \in [1 \mathinner{..} \lfloor (B^* - r - 1)/2 \rfloor], \tag{8}$$

for which we define $g_0 = 1$.

The constraints (3)-(8) with the objective of minimising K constitute the Unit Radiation model.

3.2 The Leaf-Implicit Model

This model of [2] is based on calculating an upper bound on K, denoted by $\bar{K}$. A value for $\bar{K}$ is not difficult to compute; the cardinality of any solution to the (polynomially solvable) minimum beam-on time problem will do. For each $k \in [1..\bar{K}]$, a C1 matrix X_k and associated beam-on time b_k need to be found. If X_k is the zero matrix then pattern k is not needed; minimising decomposition cardinality is minimising the number of non-zero matrices in the decomposition.

The model of [2] uses a characterisation of matrix decomposition into C1 matrices derived in [3]. In this model, the structure of the solution is encoded by recording beam-on time against each leaf position. It uses integer variables $x_{k,i,j}$ to represent the beam-on time for pattern k if the left leaf in row i of that pattern covers exactly the columns $[0 \mathinner{..} j-1]$; otherwise $x_{k,i,j}$ is zero. The left leaf being in position 0 means it is fully retracted. Similarly, the integer variable $y_{k,i,j}$ represents the beam-on time for pattern k if the right leaf in row i of that pattern covers exactly the columns $[j \mathinner{..} n+1]$, and is zero otherwise. The right leaf "covering" only column $n+1$ means it is fully retracted. For convenience, we define the function *inc* to compute the non-negative difference between two values,

$$\mathit{inc}(x, y) = \max(y - x, 0),$$

and the matrices Δ^+, Δ^- are defined by

$$\Delta^+_{i,j} = \mathit{inc}(I_{i,j-1}, I_{i,j}),$$
$$\Delta^-_{i,j} = \mathit{inc}(I_{i,j}, I_{i,j-1}),$$

for all $j \in [1..n+1]$, $i \in [1..m]$, where we take $I_{i,0} = I_{i,n+1} = 0$. Delivering the intensity matrix I is equivalent to asking that

$$\sum_{k=1}^{\bar{K}} x_{k,i,j} - w_{i,j} = \Delta^+_{i,j} \quad \text{and} \quad \sum_{k=1}^{\bar{K}} y_{k,i,j} - w_{i,j} = \Delta^-_{i,j},$$
$$\text{for all } i \in [1..m], j \in [1 \mathinner{..} n+1], \tag{9}$$

where the $w_{i,j}$ are non-negative integer variables. That the total beam-on time is B^* is ensured using integer variables b_k constrained by

$$\sum_{j=1}^{n+1} x_{k,i,j} = b_k \quad \text{and} \quad \sum_{j=1}^{n+1} y_{k,i,j} = b_k, \qquad \text{for all } k \in [1..\bar{K}], i \in [1..m], \tag{10}$$

and

$$\sum_{k=1}^{\bar{K}} b_k = B^*. \tag{11}$$

Counting the number of patterns is similarly encoded against leaf positions. The model uses binary variables $\ell_{k,i,j}$ to represent whether the left leaf in row i of pattern k covers exactly columns $[0\,..\,j-1]$, and $r_{k,i,j}$ to represent whether the right leaf in row i of pattern k covers exactly columns $[j\,..\,n+1]$. Further, binary variables β_k indicate whether pattern k is used at all. The pattern structure is enforced by the constraints

$$\sum_{j=1}^{n+1} \ell_{k,i,j} = \beta_k \quad \text{and} \quad \sum_{j=1}^{n+1} r_{k,i,j} = \beta_k, \qquad \text{for all } k \in [1..\bar{K}], i \in [1..m], \tag{12}$$

and, ensuring that the left leaf is indeed to the left of the right leaf,

$$\sum_{j=1}^{s} \ell_{k,i,j} - \sum_{j=1}^{s} r_{k,i,j} \geqslant 0, \quad \text{for all } s \in [1\,..\,n+1], k \in [1..\bar{K}], i \in [1..m]. \tag{13}$$

If pattern k is not used, then it cannot supply any radiation. This logic is encoded via the constraints

$$x_{k,i,j} \leqslant M^+_{k,i,j} \ell_{k,i,j} \quad \text{and} \quad y_{k,i,j} \leqslant M^-_{k,i,j} r_{k,i,j}, \qquad \text{for all } s \in [1..\bar{K}], i \in [1..m], j \in [1\,..\,n+1], \tag{14}$$

where $M^+_{k,i,j}$, $M^-_{k,i,j}$ are any appropriate upper bounds. We use

$$M^\circ_{k,i,j} = B^* - \sum_{s=1}^{n+1} \Delta^+_{i,s} + \Delta^\circ_{i,j}, \quad \text{for } \circ \in \{+,-\}.$$

The decomposition cardinality is found by

$$K = \sum_{k=1}^{\bar{K}} \beta_k. \tag{15}$$

The description of the original model in [2] does not discuss symmetry breaking. To make the comparison with the other models fairer, we add the following symmetry breaking constraints. In a closed row, the leaves can meet anywhere. We choose the point at which the left leaf is fully retracted, by requiring

$$\ell_{k,i,j} + r_{k,i,j} \leqslant 1, \qquad \text{for all } i \in [1..m], j \in [2 \,..\, n+1]. \tag{16}$$

Furthermore, we order the beam-times associated to the patterns,

$$b_1 \leqslant \ldots \leqslant b_{\bar{K}}. \tag{17}$$

Unfortunately, symmetry-breaking constraint (17) does not remove symmetries arising when the coefficients of two of the X_k matrices are equal. Sometimes, values can be swapped between matrices without breaking their consecutive-ones properties. Consider a fragment of D_2 from Example 1:

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

Note that the matrices of left-hand side and right-hand side are both lexicographically ordered (row-wise as well as column-wise).

In summary, the Leaf-Implicit model consists of the constraints (9)-(17) with the objective of minimising K.

4 New Constraint Programming and Integer Programming Approaches

The problem specification gives rise to a number of interesting models. Our interest is to compare models in combination with solving techniques. Although we try to model solver-independently, we need to get specific eventually, so we target integer linear programming (IP) solvers on the one hand, and constraint programming (CP) solvers (allowing arbitrary constraints) on the other.

We first discuss the most direct model that could be derived from the formulation, as a CP model. Clearly this has a number of drawbacks, such as a great deal of symmetry, so we go on to develop a compact model, useful for both IP and CP, in which much of the symmetry is eliminated.

4.1 The Direct CP Model

The problem specification can almost directly be interpreted as a CP model. The decision variables are the binary variables $X_{k,i,j}$ and the positive integer variables b_k. Requirement (1) is a linear equality constraint. Requirement (2) corresponds to the contiguity constraint studied in [12]. The critical point is that the number of variables depends on K. Hence, as in the Leaf-Implicit model, we need to make use of an upper bound $\bar{K}$ on K and program the search to try increasing values of K.

The great deal of symmetries permitted by the Direct model is a drawback. We can add (17) to remove some of the symmetries, and indeed some CP systems provide support for the combination of the constraints in (11) and (17), yielding stronger constraint propagation, e.g. the ordered_sum constraint of ECL^{i}PS^e [16]. Still, as we have seen, many symmetries remain.

4.2 The Counter Model

This novel model is based on counting the patterns according to their beam-on times. We use non-negative integer variables $Q_{b,i,j}$ to represent the *number* of patterns that have associated beam-on time b and expose the element (i, j). An upper bound $\bar{b}$ on the beam-on times is thus needed. It is easy to see that the maximum intensity, i.e. the largest value in I, is such a bound. The link between the $Q_{b,i,j}$ variables and the intensity matrix is

$$\sum_{b=1}^{\bar{b}} bQ_{b,i,j} = I_{i,j}, \qquad \text{for all } i \in [1..m], j \in [1..n]. \tag{18}$$

To derive a C1 decomposition of I from Q satisfying the above constraint, we must take a C1 decomposition of Q_b for each b. The C1 matrices in the decompositions of Q_b each have a *multiplicity* given by the number of times they occur in the decomposition of Q_b.

We claim that we can restrict our attention to decompositions of Q_b in which the multiplicities are all precisely 1. Imagine to the contrary a decomposition of I into C1 matrices $X_1, \ldots, X_K$ with respective weights $b_1, \ldots, b_K$ such that $X_i = X_j$ for some i, j with $1 \leqslant i < j \leqslant K$. Then we can construct a smaller cardinality solution, by replacing $b_i X_i + b_j X_j$ by a single C1 matrix X_i with weight $b_i + b_j$. This results in a decomposition of I with strictly smaller cardinality. Hence in any minimal cardinality decomposition of I there are no repeated C1 matrices. Hence in any minimal cardinality decomposition of I all matrices in the decompositions of Q_b have unit multiplicity.

For example, consider the following decomposition of $I = \left(\begin{smallmatrix}2&4&3\\3&4&2\end{smallmatrix}\right)$, which has non-unit multiplicity in the decomposition of Q_1:

$$1\left\{\overset{X_1}{\begin{pmatrix}0&1&0\\0&1&0\end{pmatrix}} + \overset{X_2}{\begin{pmatrix}0&1&0\\0&1&0\end{pmatrix}} + \overset{X_3}{\begin{pmatrix}0&0&1\\1&0&0\end{pmatrix}}\right\} + 2\overset{X_4}{\begin{pmatrix}1&1&1\\1&1&1\end{pmatrix}} = 1\overset{Q_1}{\begin{pmatrix}0&2&1\\1&2&0\end{pmatrix}} + 2\overset{Q_2}{\begin{pmatrix}1&1&1\\1&1&1\end{pmatrix}}.$$

Here $X_1 = X_2$, so we can replace these by a single matrix in the decomposition with weight $b_1 + b_2 = 2$. The new decomposition and the resulting Q_b matrices, Q'_1 and Q'_2, are:

$$1\overset{X_1}{\begin{pmatrix}0&0&1\\1&0&0\end{pmatrix}} + 2\left\{\overset{X_2}{\begin{pmatrix}1&1&1\\1&1&1\end{pmatrix}} + \overset{X_3}{\begin{pmatrix}0&1&0\\0&1&0\end{pmatrix}}\right\} = 1\overset{Q'_1}{\begin{pmatrix}0&0&1\\1&0&0\end{pmatrix}} + 2\overset{Q'_2}{\begin{pmatrix}1&2&1\\1&2&1\end{pmatrix}}.$$

From the above reasoning, we can assume that decompositions of Q_b into C1 matrices, each with unit weight, exist. So we have created a simpler form of the original problem. Instead of looking for a weighted decomposition of I, we seek an unweighted decomposition of each Q_b. We introduce non-negative integer variables N_b to represent the cardinality of the decomposition of Q_b, for each $b \in [1..\bar{b}]$. As we have argued, we may assume N_b is also the sum of weights for the minimum sum of weights decomposition of Q_b.

A convenient formula for the minimum sum of weights of a C1 decomposition is given in [8, 3]. The idea is as follows. The decomposition of any general non-negative integer row vector V with m elements into positive integer-weighted C1 matrices (row vectors) must satisfy the property that for all $j \in [1..m]$, the sum of weights applied to C1 matrices that expose element j but not $j-1$ must be exactly $V_j - V_{j-1}$ in the case that this is non-negative, and zero otherwise, i.e. must be exactly $inc(V_{j-1}, V_j)$, where we define $V_0 = 0$. Each nonzero C1 matrix in the decomposition must have a first element equal to one; i.e. there must be some element $j \in [1..n]$ with $j-1$ not exposed. So the sum of weights applied to nonzero C1 matrices in the decomposition must be $\sum_{j=1}^{n} inc(V_{j-1}, V_j)$. This observation extends to an $m \times n$ non-negative integer matrix G. It is straightforward to show that any decomposition of G into non-zero C1 matrices has a sum of weights equal to the maximum over $i \in [1..m]$ of

$$\sum_{j=1}^{n} inc(G_{i,j-1}, G_{i,j}) \tag{$\star$}$$

where we define $G_{i,0} = 0$ for all $i \in [1..m]$. Indeed, this quantity minimises the sum of weights over all decompositions of G into $C1$ matrices.

Thus, we can calculate N_b by finding the smallest N_b satisfying

$$N_b \geqslant \sum_{j=1}^{n} inc(Q_{b,i,j-1}, Q_{b,i,j}), \qquad \text{for all } i \in [1..m], \tag{19}$$

where we define $Q_{b,i,0} = 0$, for all b and i.

To summarise, the variables N_b represent the number of patterns that have associated beam-on time of b, and the matrix Q_b encodes the C1 matrices in the decomposition of I that should be given weight b. In other words, the matrix Q_b should itself decompose into (a sum of) N_b C1 matrices, each of which appears in the decomposition of I with weight b. Since we can restrict our attention to the decompositions of Q_b with unit multiplicities, the cardinality of the decomposition of Q_b is precisely the sum of the multiplicities. Furthermore, since we seek to minimise the cardinality of the solution, we can take N_b to be the minimum sum of multiplicities over C1 decompositions of Q_b, i.e. N_b can be related to Q_b via (19). The cardinality of a decomposition corresponding to N and Q is given by

$$K = \sum_{b=1}^{\bar{b}} N_b, \tag{20}$$

and for the total beam-on time we find

$$B^* = \sum_{b=1}^{\bar{b}} bN_b. \tag{21}$$

The Counter model thus consists of the constraints (18)-(21) with the objective of minimising K.

The Counter Model with Integer Programming. To express the Counter Model as an IP, the nonlinear constraint (19), involving max expressions, needs to be linearised. We do this by replacing the *inc* expressions in (19), that is, $\max(Q_{b,i,j} - Q_{b,i,j-1}, 0)$, by new variables $S_{b,i,j}$ constrained by

$$\begin{array}{l} S_{b,i,j} \geqslant Q_{b,i,j} - Q_{b,i,j-1}, \\ S_{b,i,j} \geqslant 0, \end{array} \quad \text{for all } b \in [1..\bar{b}], i \in [1..m], j \in [1 \mathinner{..} n+1]. \tag{22}$$

This transformation is correct since K and hence the (non-negative) N_b and $S_{b,i,j}$ are minimised.

The Counter Model with Constraint Programming. The Counter Model is directly implementable in CP systems that provide linear arithmetic constraints and the max constraint. The constraints (19) will usually be decomposed into linear inequalities over new variables representing the max expression. Our implementation uses bounds($\mathcal{R}$)-consistency for all linear arithmetic, and decomposes (19) as explained. An important part of a CP solution is the strategy used to search for a solution, which we choose as follows:

```
minimise $K$ by branch-and-bound search
    for $b := 1$ to $\bar{b}$
        instantiate $N_b$ by lower half first bisection
    $S := [1..n]$
    while $S \neq \varnothing$
        choose the row $i \in S$ with greatest row hardness
        $S := S - \{i\}$
        for $j := 1$ to $m$
            for $b := 1$ to $\bar{b}$
                instantiate $Q_{b,i,j}$ by lower half first bisection
        on failure break (return to the last choice on $N_b$)
```

After the N_b variables are fixed, rows are investigated in order of hardness. The hardness of row i is defined as the value of the expression ($\star$) with $G_{i,j} = I_{i,j}$. It captures the minimal sum of weights required to build a solution to that row.

The search strategy uses a simple form of intelligent backtracking based on the constraint graph. $Q_{b,i,j}$ and $Q_{b',i',j'}$ where $i \neq i'$ do not appear directly in any constraint together, and once the N_b are fixed the remaining constraints are effectively partitioned into independent problems on i. Hence failure for any row i indicates we must try a different solution to N_b.

While the ordering of the rows can make an order of magnitude improvement in performance, the independent solving of the subproblems is vital for tackling the larger problems.

5 Benchmarks

We tested several model/solver combinations on random intensity matrices. The parameters were their dimension, ranging from 3 × 3 to 10 × 10, and their maximum value, ranging from 3 to 15. For each parameter combination we considered

30 instances. We set a time limit of 30 minutes per instance. All benchmarks were run on the same hardware, a PC with a 2.0 GHz Intel Pentium M Processor and 2.0 GB RAM. The IP solver was CPlex version 9.13. As the CP platform we used the prototype currently being developed on top of the Mercury system [5].

We compare the Unit Radiation model, the Leaf-Implicit model, the Counter model, all with IP, and the Counter model with CP. A subset of the results are shown in Table 1. We show the average CPU times (of all times including time outs) and maximum CPU time in seconds, and in parentheses the number of instances that timed out for a parameter combination. A '—' represents that all instances timed out, and a blank entry indicates we did not run any instances since the approach was unable to effectively solve smaller instances.

Clearly the Unit Radiation model is bettered by the Leaf-Implicit model which is again substantially bettered by the Counter model. The CP solution is substantially better than the IP solution to the Counter model because of the ability to decompose the problem into independent sub-problems after the N_b are fixed.

We also experimented with some other models. The Unit Radiation and Leaf-Implicit models without symmetry breaking performed significantly worse than the models with symmetry breaking, as expected. The direct CP model described in Section 4.1 worked for very small dimensions (4,5) but did not scale; therefore, no benchmark results are reported. Finally, we experimented with a CP/IP hybrid of the Counter model, where the linear relaxation of the IP model is used as a propagator on the objective function and to check relaxed global satisfiability inside the CP search (see e.g. [14]). While the hybrid decreased the search space, and sometimes substantially so, the overhead of running the LP solver meant the resulting times were many times the pure CP solving time.

6 Concluding Remarks

We have defined the Counter model for minimal cardinality decomposition of integer matrices with the consecutive-ones property. The model significantly improves upon earlier models for the same problem, in both an integer programming and constraint programming formulation. Its critical feature is an indexing that avoids introducing symmetries.

A drawback of the Counter model is that, as in the Unit Radiation model, the number of variables depends on the maximum intensity. For the practically interesting cases in cancer radiation therapy, this may not be an issue: in instances available to us, the maximum intensity does not exceed 20. It would be interesting to see if there are other problems where the approach of indexing on number of patterns can lead to good models.

Finally, practical problem instances may have larger dimensions: current multileaf collimators allow up to 40 rows (although the outer ones may largely be empty), making further efficiency improvements useful. For example, the Counter model in a CP solver might benefit from a special constraint for (19) to avoid decomposing it into parts, where propagation strength is lost.

	Unit Radiation		**Leaf-Implicit**		**Counter/IP**		**Counter/CP**	
	CPU time (s)		CPU time (s)		CPU time (s)		CPU time (s)	
Max. val.	avg.	max.	avg.	max.	avg.	max.	avg.	max.
5 × 5								
3	33.02	844.05	2.17	55.23	0.01	0.01	0.00	0.02
4	64.98	1479.85	2.32	47.45	0.01	0.05	0.01	0.02
5	120.41	(1) 1800	2.48	7.52	0.01	0.03	0.01	0.06
6	509.14	(7) 1800	78.76	180.01	0.02	0.05	0.04	0.07
7	609.33	(9) 1800	84.89	610.70	0.02	0.07	0.05	0.07
8	845.41	(11) 1800	639.67	(8) 1800	0.05	0.26	0.06	0.08
9	728.39	(9) 1800	614.61	(10) 1800	0.06	0.28	0.07	0.09
10	1183.06	(15) 1800	797.61	(13) 1800	0.08	0.39	0.07	0.09
11	1416.51	(21) 1800	712.34	(10) 1800	0.10	0.21	0.08	0.11
12	1369.00	(19) 1800	989.09	(16) 1800	0.22	1.84	0.08	0.11
13	1596.02	(21) 1800	1341.48	(22) 1800	0.28	1.73	0.09	0.16
14	—	—	—	—	0.41	2.75	0.11	0.20
15	—	—	—	—	0.54	1.52	0.12	0.25
8 × 8								
3	1085.75	(17) 1800	731.85	(10) 1800	0.01	0.02	0.05	0.12
4	1484.24	(23) 1800	950.58	(11) 1800	0.03	0.05	0.06	0.06
5	1553.29	(23) 1800	1586.38	(22) 1800	0.06	0.09	0.06	0.08
6			—	—	3.87	45.19	0.08	0.09
7			—	—	0.51	3.96	0.09	0.11
8			—	—	133.74	(1) 1800	0.12	0.19
9			—	—	74.56	(1) 1800	0.15	0.24
10			—	—	372.53	(5) 1800	0.26	0.55
11			—	—	232.80	(2) 1800	0.39	2.07
12			—	—	507.40	(8) 1800	0.73	5.28
13			—	—	743.32	(11) 1800	0.87	2.14
14			—	—	—	—	1.36	4.19
15			—	—	—	—	2.45	6.37
10 × 10								
3					0.02	0.04	0.07	0.12
4					0.05	0.25	0.06	0.08
5					0.17	1.64	0.07	0.09
6					1.69	15.16	0.09	0.14
7					108.95	(1) 1800	0.12	0.21
8					215.97	(3) 1800	0.20	0.39
9					807.67	(12) 1800	0.46	4.51
10					1120.93	(18) 1800	0.87	4.75
11					1068.42	(14) 1800	0.97	2.82
12					1447.72	(23) 1800	1.79	7.86
13					—	—	6.84	46.89
14					—	—	15.41	133.22
15					—	—	21.17	118.51

Table 1. Benchmark results

References

1. R.K. Ahuja and H.W. Hamacher. Linear time network flow algorithm to minimize beam-on-time for unconstrained multileaf collimator problems in cancer radiation therapy. *Networks*, 45(1):36–41, 2004.
2. D. Baatar. *Matrix decomposition with time and cardinality objectives: Theory, Algorithms and Application to Multileaf collimator sequencing.* PhD thesis, University of Kaiserslautern, Germany, 2005.
3. D. Baatar, H. W. Hamacher, M. Ehrgott, and G. J. Woeginger. Decomposition of integer matrices and multileaf collimator sequencing. *Discrete Applied Mathematics*, 152(1-3):6–34, 2005.
4. N. Bansal, D. Coppersmith, and B. Schieber. Minimizing setup and beam-on times in radiation therapy. In J. Díaz, K. Jansen, J. D. P. Rolim, and U. Zwick, editors, *Proc. 9th Int. WS on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'06)*, volume 4110 of *LNCS*, pages 27–38. Springer, 2006.
5. R. Becket, M. J. Garcia de la Banda, K. Marriott, Z. Somogyi, P. J. Stuckey, and M. Wallace. Adding constraint solving to Mercury. In P. Van Hentenryck, editor, *Proc. 8th Int. Symposium of Practical Aspects of Declarative Languages (PADL'06)*, volume 3819 of *LNCS*, pages 118–133. Springer, 2006.
6. D. Z. Chen, X.S. Hu, C. Wang, and X.R. Wu. Mountain reduction, block matching, and applications in intensity-modulated radiation therapy. In *Proc. 21st Annual Symposium on Computational Geometry*, pages 35–44, 2005.
7. M. Dirkx. *Static and dynamic intensity modulation in radiotherapy using a multileaf collimator.* PhD thesis, Daniel de Hoed Cancer Centre, University Hospital Rotterdam, The Netherlands, 2000.
8. K. Engel. A new algorithm for optimal multileaf collimator leaf segmentation. *Discrete Applied Mathematics*, 152(1-3):35–51, 2005.
9. H. W. Hamacher and K.-H. Kuefer. Inverse radiation therapy planning: A multiple objective optimisation approach. *Berichte des ITWM*, 12, 1999.
10. T. Kalinowski. *Optimal multileaf collimator field segmentation.* PhD thesis, University of Rostock, Germany, 2005.
11. M. Langer, V. Thai, and L. Papiez. Improved leaf sequencing reduces segments of monitor units needed to deliver IMRT using MLC. *Medical Physics*, 28:2450–58, 2001.
12. M. J. Maher. Analysis of a global contiguity constraint. In *Proc. 4th Workshop on Rule-based Constraint Reasoning and Programming (RCoRP'02)*, 2002.
13. H. E. Romeijn, R. K. Ahuja, J. F. Dempsey, A. Kumar, and J. G. Li. A novel linear programming approach to fluence map optimization for intensity modulated radiation therapy planning. *Phys. Med. Biol.*, 48:3521–3542, 2003.
14. K. Shen and J. Schimpf. Eplex: Harnessing mathematical programming solvers for constraint logic programming. In P. van Beek, editor, *Proc. 11th Int. Conference on Principles and Practice of Constraint Programming*, pages 622–636, 2005.
15. J. E. Tepper and T. R. Mackie. Radiation therapy treatment optimization. In *Seminars in Radiation Oncology*, volume 9 of *1*, pages 1–117, 1999.
16. M. G. Wallace, S. Novello, and J. Schimpf. ECLiPSe: A platform for constraint logic programming. *ICL Systems Journal*, 12(1):159–200, 1997.
17. S. Webb. Intensity modulated radiation therapy. In *Institute of Physics Publishing Bristol and Philadelphia*, 2001.