A Network Flow Algorithm to Minimize Beam-On Time for Unconstrained Multileaf Collimator Problems in Cancer Radiation Therapy

Ravindra K. Ahuja
Department of Industrial and Systems Engineering, University of Florida, Gainesville, Florida 32611

Horst W. Hamacher
Fachbereich Mathematik, Universitaet Kaiserslautern, Kurt Schumacher Str. 26, D-67653 Kaiserslautern, Germany

In this article, we study the modulation of intensity matrices arising in cancer radiation therapy using multileaf collimators. This problem can be formulated by decomposing a given $m \times n$ integer matrix into a positive linear combination of (0, 1) matrices with the strict consecutive 1's property in rows. We consider a special case in which no technical constraints have to be taken into account. In this situation, the rows of the intensity matrix are independent of each other and the problem is equivalent to decomposing m intensity rows—independent of each other—into positive linear combinations of (0, 1) rows with the consecutive 1's property. We demonstrate that this problem can be transformed into a minimum cost flow problem in a directed network that has the following special structures: (1) the network is acyclic; (2) it is a complete graph (that is, there is an arc (i, j) whenever $i < j$); (3) each arc cost is 1; and (4) each arc is uncapacitated (that is, it has infinite capacity). We show that using this special structure, the minimum cost flow problem can be solved in $O(n)$ time. Because we need to solve m such problems, the total running time of our algorithm is $O(nm)$, which is an optimal algorithm to decompose a given $m \times n$ integer matrix into a positive linear combination of (0, 1) matrices. © 2004 Wiley Periodicals, Inc. NETWORKS, Vol. 45(1), 36–41 2005

Keywords: network flows; minimum cost flows; linear programming; shortest paths; medical applications; radiation therapy

1. INTRODUCTION

During radiation therapy, beams of radiation pass through a patient's body, depositing energy along the path of the beams. High doses of radiation kill both cancerous and normal cells. Therefore, the radiation treatment must be carefully planned so that a clinically prescribed dose is delivered to cancerous cells while sparing normal cells in the surrounding organs and tissues. To accomplish this, beams of radiation are generated by a device called a *linear accelerator* and delivered through a *beam head* at a number of different orientations, called *gantry angles*. These beams are spaced around the patient so that their intersection targets the cancerous cells, which receive the highest dose, whereas the normal cells receive radiation from some but not all the beams.

The beam head at each gantry angle can be conceived of as a rectangle that is discretized into an $m \times n$ rectangular grid; each individual rectangle in this grid is called a *bixel*. At each gantry angle, the radiation head delivers the radiation dose, which is specified by an $m \times n$ matrix $\mathbf{M}$, called the *intensity matrix*. We assume $\mathbf{M}$ to be an integer valued matrix. For example, if we discretize the beam head into a 5×4 grid, then one possible intensity matrix is:

$$\mathbf{M} = \{\mathbf{M}_{ij}\} = \begin{bmatrix} 4 & 4 & 3 & 0 \\ 1 & 6 & 3 & 0 \\ 3 & 4 & 1 & 0 \\ 4 & 4 & 3 & 0 \\ 3 & 6 & 4 & 3 \end{bmatrix} \quad (1)$$

However, the radiation generated in the linear accelerator provides uniform intensity for all bixels. To generate the nonuniform intensity matrix from a uniform intensity source, a device called a *multileaf collimator* (MLC) is used. A MLC has m rows, called *channels*, and each row has a *left leaf* and a *right leaf*, whose positions can be changed; the radiation can pass in between the left and right leaves. If $\mathbf{M}$ has n columns $1, 2, \ldots, n$, then for each row i, there are $n + 1$ positions, $1, 2, \ldots, n, n + 1$, at which the left and right leaves can be positioned (see Fig. 1). If the left leaf is at position l and the right leaf is at position r, then

Received June 2003; accepted October 2004.
Correspondence to: R. K. Ahuja; e-mail: ahuja@ufl.edu
Contract grant sponsor: National Science Foundation; Contract grant numbers: DMI-990087 and DMI-0217359 (to R.K.A.).
Contract grant sponsor: German Cancer Association (to H.W.H.).
Contract grant sponsor: German Research Council (Deutsche Forschungsgemeinschaft); contract grant number: 1785/7-2 (to H.W.H.).
DOI 10.1002/net.20047
Published online in Wiley InterScience (www.interscience.wiley.com).

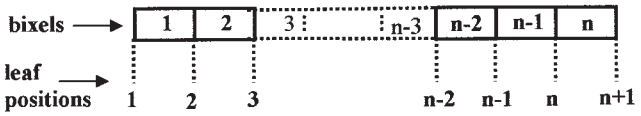

FIG. 1. A single channel (row) of a multileaf collimator with the left leaf at position $l = 3$ and the right leaf at position $r = n - 2$. Radiation passes through the bixels $3, \ldots, n - 3$.

the radiation will pass through the bixels numbered l, $l + 1, \ldots, r - 1$.

Choices of the left and right leaves in all rows can be specified by a 0-1 matrix called a *shape matrix*. A "0" in the shape matrix indicates the corresponding bixel is blocked by the leaves and does not deliver any radiation, and a "1" indicates that the corresponding bixel delivers the radiation. Figure 2 presents some possible shape matrices. Observe that each shape matrix satisfies the *consecutive* 1's *property* that all the 1's in each row are consecutive.

To deliver the given intensity matrix $\mathbf{M}$, the linear accelerator sends a uniform beam of radiation through the MLCs with different shape matrices $S_1, S_2, \ldots, S_K$ for different lengths of time $x_1, x_2, \ldots, x_K$, called *beam-on times*, such that $\mathbf{M} = \sum_{k=1}^{K} S_k x_k$. In Figure 3, we illustrate a realization of a given intensity matrix through three shape matrices.

The total delivery time is the sum of the beam-on times plus the setup times needed to go from one shape matrix to another shape matrix. Let $c(S_k, S_{k+1})$ denote the setup time needed to go from the shape matrix S_k to the shape matrix S_{k+1}. In the above example, we need three setups and the beam-on time is $3 + 1 + 2 = 6$ units. The *minimum delivery time problem* is to determine the shape matrices $S_1, S_2, \ldots, S_K$ and their corresponding beam-on times so as to

$$\text{minimize } z^* := \sum_{k=1}^{K} x_k + \sum_{k=0}^{K-1} c(S_k, S_{k+1}) \tag{2a}$$

subject to

$$\sum_{k=1}^{K} S_k x_k = \mathbf{M} \tag{2b}$$

S_k is a valid shape matrix and $x_k > 0$ for all k

$$= 1, 2, \ldots, K, \tag{2c}$$

where S_0 is the initial setup matrix, which can correspond to

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

FIG. 2. Examples of shape matrices.

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

FIG. 3. Realizing an intensity matrix through three shape matrices.

the MLC completely closed. Note that in this formulation all variables are required to be strictly positive because setup costs only have to be considered for those variables. In particular, K, the number of shape matrices used in the decomposition is part of the output of the problem. This becomes obvious in the special case of the minimum delivery time problem when we assume that the setup time is constant, that is, independent of the shape matrices. We refer to this problem as the *minimum delivery time problem with constant setup times*. Burkard [7] has demonstrated that this problem is NP-hard (see Section 3). Therefore, the general problem and its special case are unlikely to be solvable in polynomial time. We will thus focus on a special case of (2) for which setup times are assumed to be negligible, compared to the beam-on times, and so can be ignored. This problem, referred to in the following as the *minimum beam-on time problem*, is

$$\text{minimize } z^* := \sum_{k=1}^{L} x_k \tag{3a}$$

subject to

$$\sum_{k=1}^{L} S_k x_k = \mathbf{M} \tag{3b}$$

S_k is a valid shape matrix and $x_k \geq 0$ for all k

$$= 1, 2, \ldots, L. \tag{3c}$$

Note that, in contrast to Problem (2), the variables are no longer required to be strictly greater than 0 because we take the sums in the objective function and the constraints over the set of all possible shape matrices indexed by $1, 2, \ldots, L$. Therefore, L in (3) is the total number of shape matrices.

To be more specific, we should refer to this problem as the *unconstrained* minimum beam-on time problem because we do not consider any technological constraints on the multileaf collimator. The same approach has been taken by Bortfeld et al. [6] in their sweep algorithm. In fact, our article can be considered an efficient implementation of the sweep algorithm and a rigorous proof that this technique does indeed provide an optimal solution.

Depending on the technology of the multileaf collimator, constraints may have to be taken into account. Often, for instance, adjacent interleaf pairs cannot overlap. To include this constraint in the model, we can add *interleaf motion constraints*, which state that the left leaf in channel i must

be left of the right leaves of channels $(i - 1)$ and $(i + 1)$, and that the right leaf in channel i must be right of the left leaves in channels $(i - 1)$ and $(i + 1)$. The resulting constrained beam-on time problem is obviously more difficult. It has been tackled by Lenzen [9] and Hamacher and Lenzen [8] using integer programming models and by Boland et al. [5] using a network flow approach. In the latter article, a polynomial-time algorithm solves the minimum beam-on time to optimality. It uses, however, a large network with $O(mn^2)$ nodes, so that problems with a large number of bixels in the multileaf collimator cannot be solved within a reasonable amount of time. The article also contains a discussion of other articles dealing with multileaf collimator problems. Out of these, the heuristics of Xia and Verhey [12] and of Siocci [11] are probably the best known. The latter heuristic has been implemented as the IMFAST procedure, and is part of the commercial radiation system CORVUS. More recent articles by Baatar [2] and Baatar and Hamacher [3] reduced the number of variables in the Boland et al. [5] model by a factor of n and thus improved the computational time considerably.

The fact that our article deals with the unconstrained problem is justified by the technology of some MLCs and the possibility of using any optimal solution of (3) as a starting point of a more involved optimization model. Preciado-Wolters et al. [10] reveal that this approach is useful in an attempt to design an integrated cancer radiation decision support system in which the question of implementing a given intensity matrix by multileaf collimators is only one of several other optimization problems that need to be solved.

Because the problem does not have any constraints, the rows of the intensity matrix can be treated independently, that is, the problem is equivalent to m independent problems of decomposing an intensity row into a positive linear combination of (0, 1) rows with the consecutive 1's property. We demonstrate that the minimum beam-on time problem for each row can be transformed into a minimum cost flow problem in a directed network that has the following special structures: (1) the network is acyclic (it contains no cycles); (2) the network is complete [there is an arc (i, j) whenever $i < j$]; (3) each arc cost is 1; and (4) each arc is uncapacitated (it has infinite capacity). We show that by using this special structure, the minimum cost flow problem can be solved in $O(n)$ time. Because we need to solve m such problems, the total running time of our algorithm is $O(nm)$, which is an optimal algorithm to decompose a given $m \times n$ integer matrix into a positive linear combination of (0, 1) matrices.

2. THE MINIMUM COST FLOW FORMULATION

The minimum beam-on time problem for the ith row of the intensity matrix can be stated as

$$\text{minimize } z_i := \sum_{k=1}^{K'} x_{ik} \tag{4a}$$

subject to

$$\sum_{k=1}^{K'} \mathbf{R}_k x_{ik} = b_i \tag{4b}$$

$$x_{ik} \geq 0 \text{ for all } k = 1, 2, \ldots, K'. \tag{4c}$$

Here, b_i denotes the ith row of the intensity matrix $\mathbf{M}$, and $\{R_k: 1 \leq k \leq K'\}$ denotes the set of all possible shape row vectors. In this formulation, the x_{ik}s are decision variables that denote the time during which the shape row R_k is exposed to radiation. The following lemma relates the optimal solution of (3) with the optimal solution of (4):

Lemma 1. *Let* $z_{max} = max\{z_i: 1 \leq i \leq m\}$. *Then,* $z^* = z_{max}$ *is the optimal objective function value of* (3).

Proof. It is evident that each z_i is a lower bound on z^*, the optimal objective function value of (3). Therefore, $z_{\max}$ is a valid lower bound on z^*. We will now demonstrate that we can construct a set of shape matrices for which the beam-on time equals $z_{\max}$; hence, $z_{\max}$ is a valid upper bound on z^*. It then follows that $z_{\max} = z^*$.

Let $W_i = \{W_{i_1}, W_{i_2}, \ldots, W_{i_p}\}$ denote the shape rows in an optimal solution for the ith row of the intensity matrix $\mathbf{M}$ with beam-on times as $x_{i_1}, x_{i_2}, \ldots, x_{i_p}$, respectively. By definition, $z_i = \sum_{q=1}^{p} x_{i_q}$. We will use the following iterative process to construct optimal shape matrices from these shape rows. At each iteration, we select a shape row, say $W_{i_{j1}}$, from the set W_i, for each i, $1 \leq i \leq m$. Putting together these shape rows gives us a shape matrix S_1. We set the beam-on time, x_1, of this shape matrix equal to the minimum of the beam-on times of the shape rows selected, that is, $x_1 = \min\{x_{i_{j1}}: 1 \leq i \leq m\}$. We next reduce the beam-on times of the selected shape matrices by x_1, and if in a shape row, say, $W_{i_{j1}}$, the beam-on time becomes zero, then we delete it from W_i. In the next iteration, we again select a shape row, $W_{i_{j2}}$, from each set W_i, $1 \leq i \leq m$ and combine these rows to create another shape matrix S_2 with beam-on time x_2 as $x_2 = \min\{x_{i_{j2}}: 1 \leq i \leq m\}$. We repeat this process until each W_i is empty. Because in the kth iteration we reduce the beam-on time of a shape row in each W_i by a constant number x_k (as long as W_i is nonempty), and we create a shape matrix with a beam-on time equal to x_k, the sum of the beam-on times of the shape matrices constructed will be equal to the maximum of all z_is, which is $z_{\max}$. This completes the proof of the lemma. ∎

We will henceforth focus on solving (4). For the sake of simplicity, we first eliminate the subscript i (representing the ith row). We will thus refer to x_{ik} as x_k and to b_i as b.

For reasons that will become clear later on, we also convert the row vector R_ks and b into column vectors by taking their transpose and, for simplicity's sake, use the same notation to represent the corresponding column vectors.

Each (column) vector $\mathbf{R}_k$ is a 0-1 vector and corresponds to a feasible (nonzero) exposure provided by a pair of left and right leaves. The feasibility of the exposure requires that all the 1's in each $\mathbf{R}_k$ consists of a (possibly null) sequence of 0s, followed by a sequence of 1's, followed by another (possibility null) sequence of 0s. We will henceforth refer to the vector $\mathbf{R}_k$ as $\mathbf{R}_{uv}$ if, in the vector $\mathbf{R}_k$, row u is the least index row with element 1 and $v-1$ is the highest index row with element 1. Let $\{\mathbf{R}_{uv}: (u, v) \in A\}$ denote the set of all possible column vectors with nonzero exposure. Because each column vector $\mathbf{R}_{uv}$ corresponds to a feasible pair of left and right leaves, and because the left leaf can take the positions $u = 1, 2, 3, \ldots, n$, and the right leaf can take the positions $v = u+1, u+2, \ldots, n+1$, it follows that $A = \{(u, v): 1 \le u \le n, u+1 \le v \le n+1\}$. Observe that $|A| = n + (n-1) + (n-2) + \cdots + 1 = n(n+1)/2$. We can now restate (4) as:

$$\text{minimize } z^0 := \sum_{(u,v)\in A} x_{uv} \tag{5a}$$

subject to

$$\sum_{(u,v)\in A} \mathbf{R}_{uv} x_{uv} = b \tag{5b}$$

$$x_{uv} \ge 0, \text{ for all } (u, v) \in A. \tag{5c}$$

We illustrate the formulation (5) using a numerical example. Suppose that $n = 3$. In this case, $A = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$ and the formulation (5) becomes:

$$\text{minimize } z^0 := x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} \tag{6a}$$

subject to

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{34} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \tag{6b}$$

$$x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34} \ge 0. \tag{6c}$$

Here, we have added a redundant zero row ($0x = 0$) to the constraints (6b), which will be used later to transform (6) to a network flow problem. Now, observe that each column $\mathbf{R}_{uv}$ has consecutive 1's in the rows $u, u+1, \ldots, v-1$. A linear program in which each column vector is a vector of 0s and 1's and all the 1's are consecutive is called a *linear program* (*LP*) *with consecutive 1's in columns* and can be transformed into a minimum cost flow problem (Ahuja et al. [1], Section 9.2). This transformation entails adding a row of zeros ($n+1$th row) to the constraint matrix (as we have done above) and subtracting each row u from the row $(u+1)$ for each $u = n, n-1, \ldots, 1$, in this stated order. These row operations provide an equivalent LP where each new column $\mathbf{R}'_{uv}$ has one $+1$ in the uth row, one -1 in the vth row, and all other values are zero. Now,

$$\text{minimize } z^0 := \sum_{(u,v)\in A} x_{uv} \tag{7a}$$

subject to

$$\sum_{(u,v)\in A} \mathbf{R}'_{uv} x_{uv} = b' \tag{7b}$$

$$x_{uv} \ge 0, \text{ for all } (u, v) \in A \tag{7c}$$

is the modified linear program. For example, if we apply these row operations to (6), we get the following equivalent system:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{34} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - b_1 \\ b_3 - b_2 \\ -b_3 \end{bmatrix} \tag{8}$$

Evidently, $\mathbf{R}'_{uv}$ has $+1$ in the uth row and -1 in the vth row. The following properties easily follow because $b_i \ge 0$ for all i, $1 \le i \le n+1$; and $b'_i = b_i - b_{i-1}$ for all i, $2 \le i \le n+1$.

Property 1. $\sum_{i=1}^{n+1} b'_i = 0$.

Property 2. $\sum_{i=1}^{j} b'_i \ge 0$ *for all* j, $1 \le j \le n+1$.

It is well known that a linear programming problem in which each column has one $+1$ and one -1, with the rest of the elements zero is a minimum cost flow problem (see, e.g., Ahuja et al. [1], Section 1.2). The minimum cost flow problem is one of the fundamental network flow problems and can be solved efficiently. Its feasibility is guaranteed by Property 1. Let $G = (N, A)$ denote the underlying network where $N = \{1, 2, 3, \ldots, n+1\}$ denotes the node set and $A = \{(u, v): u = 1, 2, \ldots, n$, and $v = u+1, u+2, \ldots, n+1\}$ denotes the arc set. In the minimum cost flow problem, the cost of the flow of each arc equals 1. For example, the LP (8) is a minimum cost flow problem in the network shown in Figure 4. In this figure, the number next to each node indicates the supply/demand of the node, and the number next to each arc denotes its cost.

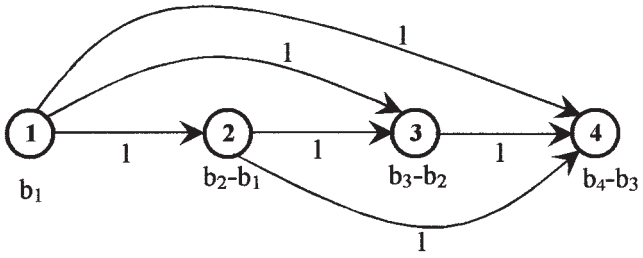

FIG. 4. The minimum cost flow formulation of the LP in (8).

The minimum cost flow problem (7) has a very special structure allowing a very efficient procedure based on the following properties:

Property 3.

1. *the network is acyclic;*
2. *the network is complete (that is, it contains an arc (i, j) for every i < j);*
3. *each arc cost is 1; and*
4. *each arc is uncapacitated (that is, it has no capacity restriction).*

These properties allow us to solve the minimum cost flow problem in O(n) time using the well-known successive shortest path problem for the minimum cost flow problem (see Ahuja et al. [1], Section 9.7, for a description of this algorithm). The successive shortest path algorithm starts with a zero flow and, at each iteration, augments flow from a supply node u (with supply $b'_u > 0$) to a demand node v (with demand $b'_v < 0$) along a shortest path in the residual network until all node supplies/demands are satisfied. The residual network is defined with respect to a flow x and is denoted by $G(x)$. To construct the residual network, we replace each arc $(i, j) \in A$ by two arcs, a *forward arc* (i, j) and a *reverse arc* (j, i). The arc (i, j) has cost c_{ij} and *residual capacity* $r_{ij} = u_{ij} - x_{ij}$, and the arc (j, i) has cost $c_{ji} = -c_{ij}$ and residual capacity $r_{ji} = x_{ij}$. The residual network consists *only* of arcs with positive residual capacity. The special structure of the minimum cost flow problem (7) allows us to simplify the steps of the successive shortest path algorithm and turn it into a very simple and efficient algorithm. We describe the resulting algorithm in Figure 5 followed by its justification.

During the execution of the algorithm, we call a node u in N an *excess node* if $e(u) > 0$, and we call it a *deficit node* if $e(v) < 0$. Initially, each supply node is an excess node, and each demand node is a deficit node. The algorithm always selects node u as the least index excess node and node v as the least index deficit node. It follows from Property 2 that $u < v$, and it follows from Property 3 that arc $(u, v) \in A$. Observe that arc (u, v) is the shortest path from node u to node v using forward arcs in the residual network because each forward arc has cost 1 and any path must contain at least one arc. The backward arcs in the residual network have cost -1. But each backward arc is an

```
algorithm min-cost-flow;
begin
    for all nodes u ∈ N := {1, 2, , …, n, n+1} do e(u) := b'_u;
    u := min{r : e(r) > 0};
    v := min{r : e(r) < 0};
    while both u and v are defined do
    begin
        δ := min{e(u), -e(v)};
        x_uv := δ;
        e(u) := e(u) - δ;
        e(v) := e(v) + δ;
        while e(u) = 0 do increment u;
        while e(v) = 0 do increment v;
    end;
    return the solution x;
end;
```

FIG. 5. The algorithm for solving the minimum cost problem in (7).

incoming arc into the supply node from where augmentations are being performed and such an arc cannot lie on the shortest path. Hence, when identifying shortest paths, we can restrict attention to only the forward arcs (that is, arcs in A) which is what the algorithm described in Figure 5 does. At any iteration, the algorithm sends $\delta = \min\{e(u), -e(v)\}$ units of flow on the arc (u, v), which is a shortest path from node u to node v. This flow augmentation either reduces $e(u)$ to zero or $e(v)$ to zero. In the former case we update u, and in the later case we update v. To update u(or v), we simply increment u(or v) by 1 repeatedly, and stop when $e(u) > 0$ (or $e(v) < 0$). The total time taken to update u and v over the entire algorithm is O(n). The other steps of the algorithm also require a total of O(n) time. Therefore, the following theorem is proved.

Theorem 1. *The minimum cost flow problem* (7) *can be solved in O*(n) *time. The* (*unconstrained*) *minimum beam-on time problem can therefore be solved in O*(nm) *time.*

3. CONCLUSIONS

We have demonstrated that the (unconstrained) minimum beam-on time problem can be solved in O(nm) time. This approach is appropriate in situations when this problem has to be solved numerous times, for instance, as a subroutine in algorithms for more involved multileaf collimator models. (See, i.e., Preciado-Wolters et al. [10].) In these methods, to take technical constraints into account, more involved network flow and integer programming methods are required. (See, i.e., Boland et al. [5], and Baatar and Hamacher [3].)

We will address several additional interesting research problems in the future. Suppose that we want to minimize the delivery time (the beam-on time plus the sum of the setup times for the multileaf collimators). It was an open question for quite some time whether this problem could be solved in polynomial time. Burkard [7] established that the

problem of finding a decomposition of the intensity matrix into the smallest number of shape matrices is NP-hard, thus answering this question in the negative (unless $P = \text{NP}$). His reduction is from the subset sum problem using the following result:

Theorem 2 (Burkard [7]). *Given n numbers* $a_1, \ldots, a_n$. *Then the* $(2 \times (2n - 1))$ *matrix*

$$A = \begin{pmatrix} a_1 & 0 & a_2 & 0 & a_3 & \cdots & a_n \\ M & M & M & M & M & \cdots & M \end{pmatrix}$$

has a decomposition into n shape matrices if and only if there exists a subset $\{i_1, i_2, \ldots, i_k\}$ *of* $\{1, \ldots, n\}$ *such that* $a_{i_1} + \cdots + a_{i_k} = M$.

Baatar et al. [4] strengthened this result by showing that the problem of decomposing a single-row matrix is NP-hard in the strong sense. Because the delivery time is important, both for medical and economical reasons, algorithms for finding minimum delivery times have to be found, despite the fact that the problem is NP-hard.

Acknowledgments

We are grateful for the constructive remarks of two anonymous referees. We are also thankful to Doug Shier for carefully proofreading the manuscript and identifying some errors.

REFERENCES

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, Network flows: Theory, algorithms, and applications, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] D. Baatar, New linear programming models for multileaf collimators in radiation therapy, Master's thesis, Department of Mathematics, University of Kaiserslautern, Germany, 2002.

[3] D. Baatar and H.W. Hamacher, Fast linear programming models for constrained multileaf collimators in radiation therapy, Technical Report, Department of Mathematics, University of Kaiserslautern, Germany, 2003.

[4] D. Baatar, M. Ehrgott, H.W. Hamacher, and G. Woeginger, Decomposition of integer matrices into consecutive ones matrices and application, Research Report, Department of Mathematics, University of Kaiserslautern, Germany, 2004.

[5] N.H. 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] R.E. Burkard, Open research question section of Oberwolfach conference, November 2002.

[8] H.W. Hamacher and F. Lenzen, "A mixed integer programming approach to the multileaf collimator problem," The use of computers in radiation therapy, W. Schlegel and T. Bortfeld (Editors), Springer-Verlag, Berlin, 2000, pp. 210–212.

[9] F. Lenzen, An integer programming approach to the multileaf collimator problem, Master's Thesis, Department of Mathematics, University of Kaiserslautern, Germany, 2000.

[10] F.R. Preciado-Wolters, R. Rardin, M. Langer, and V. Thai, Column generation for IMRT cancer therapy optimization with implementable segments, Ann Operat Res, 2004, submitted.

[11] R.A.C. Siochi, Minimizing static intensity modulation delivery time using an intensity solid paradigm, Int J Radiat Oncol Biol Phys 43 (1999), 671–689.

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