INSTITUTE OF PHYSICS PUBLISHING PHYSICS IN MEDICINE AND BIOLOGY

Phys. Med. Biol. **51** (2006) 2517–2536 doi:10.1088/0031-9155/51/10/011

A difference-matrix metaheuristic for intensity map segmentation in step-and-shoot IMRT delivery

Athula D A Gunawardena[1,2], **Warren D D'Souza**[3], **Laura D Goadrich**[4], **Robert R Meyer**[2], **Kelly J Sorensen**[2], **Shahid A Naqvi**[3] **and Leyuan Shi**[4]

[1] Department of Mathematics and Computer Sciences, University of Wisconsin-Whitewater, 800 West Main Street, Whitewater, WI, USA
[2] Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI, USA
[3] Department of Radiation Oncology, School of Medicine, University of Maryland, 22 South Greene Street, Baltimore, MD, USA
[4] Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI, USA

Received 19 April 2005, in final form 4 December 2005
Published 27 April 2006

Abstract

At an intermediate stage of radiation treatment planning for IMRT, most commercial treatment planning systems for IMRT generate intensity maps that describe the grid of beamlet intensities for each beam angle. Intensity map segmentation of the matrix of individual beamlet intensities into a set of MLC apertures and corresponding intensities is then required in order to produce an actual radiation delivery plan for clinical use. Mathematically, this is a very difficult combinatorial optimization problem, especially when mechanical limitations of the MLC lead to many constraints on aperture shape, and setup times for apertures make the number of apertures an important factor in overall treatment time. We have developed, implemented and tested on clinical cases a metaheuristic (that is, a method that provides a framework to guide the repeated application of another heuristic) that efficiently generates very high-quality (low aperture number) segmentations. Our computational results demonstrate that the number of beam apertures and monitor units in the treatment plans resulting from our approach is significantly smaller than the corresponding values for treatment plans generated by the heuristics embedded in a widely use commercial system. We also contrast the excellent results of our fast and robust metaheuristic with results from an 'exact' method, branch-and-cut, which attempts to construct optimal solutions, but, within clinically acceptable time limits, generally fails to produce good solutions, especially for intensity maps with more than five intensity levels. Finally, we show that in no instance is there a clinically significant change of quality associated with our more efficient plans.

0031-9155/06/102517+20$30.00

1. Introduction

Treatment planning in IMRT usually involves a two-step process in which (1) the dose within the tumour and critical structures is first optimized without consideration of the delivery specifics and (2) the resulting intensity map (*I*-map) for each beam orientation is then segmented to produce segments (apertures) by taking into account the physical delivery constraints of the MLC. When segmenting the intensity map, one must consider both the number of shapes and the number of MUs. There are several MLCs currently in clinical use, with rather different physical constraints that determine the allowable leaf positions (and, hence, corresponding shape matrices) that are possible for that device. These machine types also have widely varying setup times for transitions between shapes, and for certain machine types these setup times can dominate the radiation delivery time, a factor that we will consider below. At our institution, IMRT treatments are planned and delivered using the Corvus, version 4.0 (Nomos Corporation, Cranberry Township, PA) treatment planning system and Elekta SL 20 (Elekta Oncology Systems, Crawley, UK) linear accelerators. The leaf-sequencing algorithm in the Corvus planning system tends to produce a large number of beam delivery segments for each beam orientation (20–70) depending on the number of modulation intensity levels and the complexity of the case. There exists an inter-segment delay of approximately 6 s on an Elekta SL linear accelerator. Coupled with a large number of delivery segments, this can significantly increase the treatment time to approximately 45 min or longer. This has the potential to increase patient discomfort and spurious intra-fraction patient motion as well as decrease clinical throughput.

The leaf-sequencing or segmentation problem described above is a particularly difficult combinatorial problem for which a variety of heuristic and ‘exact’ approaches have been proposed. Boyer and Yu (1999), Evans *et al* (1997), Galvin *et al* (1993), Bortfeld *et al* (1994), Xia and Verhey (1998) and Siochi (1999) discuss several heuristic methods that have been developed for this problem. Based on computational comparisons in Que (1999), the Xia–Verhey approach, which focuses on decompositions based on intensities that are powers of 2, performs relatively well with respect to these alternatives in terms of number of segments. On the other hand, the Bortfeld *et al* approach tends to yield larger numbers of segments, but smaller beam-on times. (In this context, beam-on time has an algebraic definition that is presented below; it is possible to map beam-on time values to monitor unit values by using an appropriate proportionality factor for each beam angle.) For the case in which only beam-on time is considered, an interesting exact network flow approach to segmentation was developed recently by Boland *et al* (2004). Luan *et al* (2004) also investigated the use of network flow techniques, by focusing on minimizing aperture count. Taking into account both aperture count and beam-on time, Langer *et al* (2001) derived an exact approach based on the solution via branch-and-cut of an integer programming model and compared their results with those of the heuristic methods of Bortfeld *et al* (1994), Boyer and Yu (1999) and Xia and Verhey (1998). Since Langer *et al*’s method is based on the ‘exact’ procedure of branch-and-cut, in theory it yields the *minimum number of segments* subject to certain delivery constraints, including a minimum beam-on time constraint. However, this assumes sufficient computing time, which we demonstrate in the results section, may be more than the time available for treatment planning. For the problem instances that they consider, Langer *et al* are able to demonstrate significant improvements relative to heuristic approaches. However, the problem of minimizing aperture counts is significantly harder than the beam-on time problem, because the former involves binary-valued step functions that have value 1 for a sub-aperture that is used (with positive radiation intensity) and value 0 otherwise. The efficient metaheuristic that we develop here coordinates the search processes arising from the use of multiple starting points for a base heuristic (which is related to a method of Engel (2003) but takes into account complex delivery constraints) and yields segmentations with small numbers of apertures and small beam-on times.

Results with our metaheuristic are compared with leaf segments produced by the heuristics included in the Corvus treatment planning system (versions 4.0 and 5.0). Additionally, we benchmark our approach against a branch-and-cut-based exact approach described by Langer *et al*. Further, we compare the resulting 3D dose distributions resulting from our metaheuristic-generated leaf sequences to the original dose distributions using an in-house developed and previously described dose calculation engine.

2. Methods and materials

For IMRT delivery, the matrix of relative beamlet intensities (generated for each beam angle by dose optimization) must be decomposed into a integer-weighted sum of binary matrices, each of which must obey a set of physical constraints dictated by the MLC (details of these constraints are discussed below). Mathematically, the constraints of the segmentation problem may be expressed as follows: given a matrix A of relative beamlet intensities (see figure 13 in appendix A for an idealized example of an I-map), determine a set of binary shape matrices S_k (satisfying the constraints of the collimator) and corresponding positive beam weights α_k such that $A = \sum_k \alpha_k S_k$. The focus of this research is to develop efficient segmentation methods that require approximately the minimum number of shapes and use small total beam-on times (where beam-on time is defined as $\sum_k \alpha_k$). Beam-on time as defined in this manner is a good surrogate for monitor units (MUs) as we will see below.

2.1. The DMM metaheuristic for intensity map segmentation

Our research has focused on the development of a *metaheuristic*, the difference-matrix metaheuristic (DMM), that coordinates repeated applications of a *base heuristic* (bDif). Roughly speaking this metaheuristic combines multi-start (the use of multiple carefully selected 'starting points' for bDif) with local search and coordination mechanisms that focus the search of the heuristic in more promising areas of the solution space as these are determined by comparisons of results obtained from the multiple start points. In this section, we first describe the base heuristic bDif and then we discuss how bDif is exploited via the multi-start and coordination aspects of the DMM metaheuristic.

The base heuristic bDif rapidly processes an input I-map by repeatedly extracting (subtracting) 'feasible' aperture–intensity pairs until the I-map has been completely segmented. (See appendix A for an example of an extraction step.) Apertures for the pairs are constructed row-by-row to satisfy Elekta delivery constraints. A matrix resulting from the pair extraction process is also an I-map (i.e., an array of non-negative integers) and will be referred to as a residual I-map. In the first major iteration, the input I-map is processed as described below by bDif. In subsequent iterations, residual I-maps are processed. Once a residual I-map contains only 0 entries, the segmentation process has been completed.

Additional criteria used by bDif in shape construction in each extraction step are based on the elements of the 'difference matrix' derived from the current I-map. (Previous research in this area (Engel 2003) has laid the theoretical groundwork for the difference-matrix viewpoint by demonstrating that these methods can produce minimum beam-on time segmentations in the absence of leaf collision constraints.) The difference matrix with respect to a given matrix consists of 'forward' differences of entries in successive columns. From a geometric viewpoint, the entries in a given row of an I-map can be thought of as heights 'landscape'

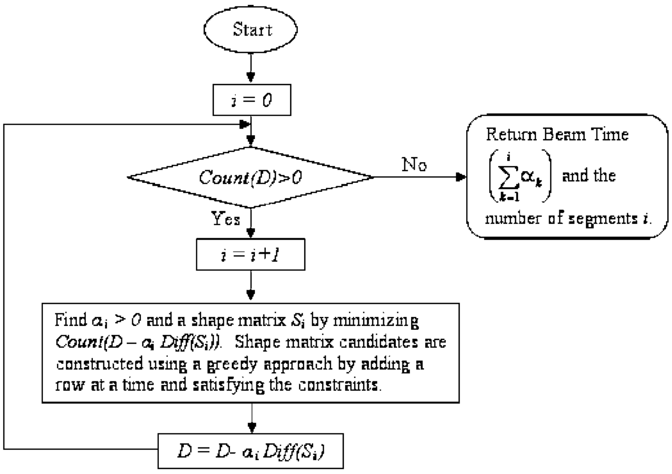


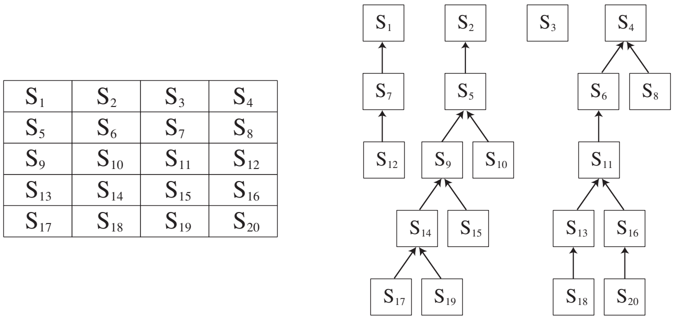
Figure 1. The main procedures of the base heuristic bDiff.

Table 1. Input and output of the DMM procedure.

Input:	an $m \times n$ intensity matrix $A = (a_{i,j})$ comprised of non-negative integers (the column indices of A are given by $1, \ldots, n$)
Output:	K binary aperture shape matrices S_k and positive integers α_k ($k = 1, \ldots, K$) giving corresponding beam-on times for the apertures. The apertures obey the delivery constraints of the Elekta MLC and the weight–shape pairs satisfy $\sum_{k=1}^{K} \alpha_k S_k = A$

relative to a baseline of 0, and thus the difference matrix also serves to represent 'steps' up (positive entries) as the landscape is traversed from left to right, steps down (negative entries) and level regions (0 entries) within this landscape. Further details and an example are given in appendix A. The Diff operator is used to denote the difference-matrix process. In the algorithmic descriptions below, D will denote a generic difference matrix that corresponds to either Diff(A), where A is the original intensity map, or to a difference matrix obtained by applying Diff to a residual intensity map. Using this notation, the input and output of the overall DMM procedure are seen in table 1.

Mechanical aspects of the Elekta MLC determine conditions that shape matrices S_k must satisfy in order to be delivered. In addition to the obvious condition that the set of open bixels in a row be connected, these constraints include (1) interleaf collision, (2) minimum leaf separation, (3) shape connectedness and (4) 'straight-edge' constraints related to back-up diaphragms. Our base heuristic bDif (figure 1) enforces these constraints by checking deliverability as each row is added to the current shape. (The row-by-row delivery constraint verification used in bDif is much more efficient than the use of the algebraic descriptions (via constraints that use binary variables) of these delivery constraints that are needed in the branch-and-cut approach, since the algebraic constraints must cover all possible cases and hence are quite numerous.) Tongue-and-groove (TnG) effects, which correspond to inter-shape rather than intra-shape properties, could be addressed in our metaheuristic by modifying it to incorporate the use of penalties for segmentations that would significantly display such effects. However, while TnG effects do occur in the segmentations produced by DMM, the dosimetric impact of these effects is not clinically significant in the cases that we have studied.



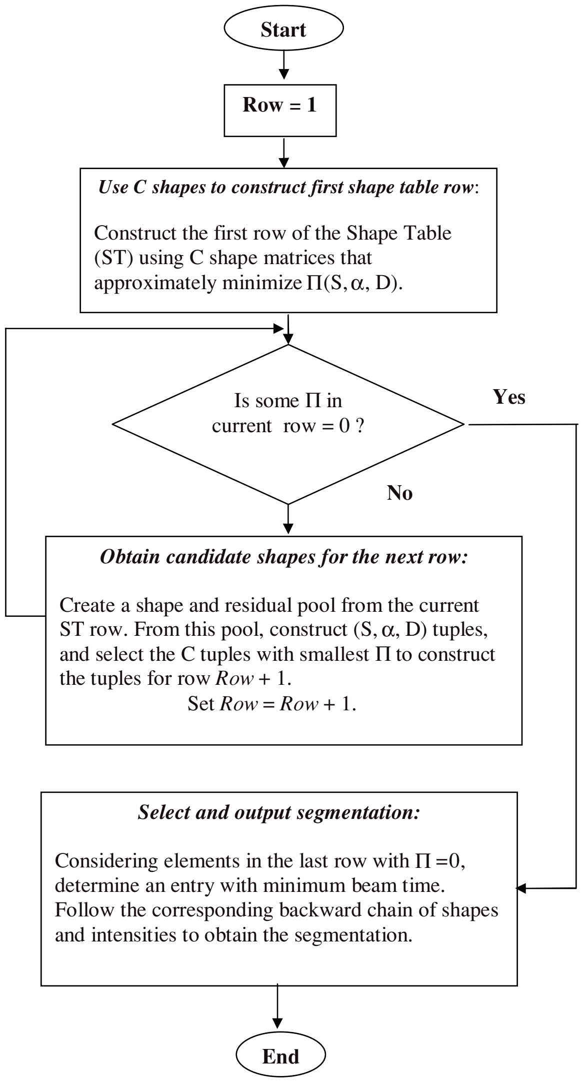
S_1	S_2	S_3	S_4
S_5	S_6	S_7	S_8
S_9	S_{10}	S_{11}	S_{12}
S_{13}	S_{14}	S_{15}	S_{16}
S_{17}	S_{18}	S_{19}	S_{20}

Figure 2. An example of a DMM multi-tree search and its corresponding shape table.

An important underlying concept of bDif is that clusters of 0 entries in a difference matrix correspond to ‘flat’ portions of the corresponding intensity map and hence suggest good shapes. Thus, bDif focuses on the construction of shapes whose extractions produce small values of the Count function ($\mathrm{Count}(D) =$ the number of non-zero entries in D). The choice of the intensity values α used in the bDif segmentation process is determined in part by considering values of α that appear in the difference matrix (see appendix B).

We now describe the metaheuristic DMM that we developed in order to have a broad-based search for a good segmentation. DMM (as indicated in the flow chart, figure 3) employs bDif within a procedure that generates (via bDif) C (a parameter discussed below) multiple starting points (see figures 2 and 3). DMM then coordinates the corresponding multiple local searches by comparing results and focusing further search effort on promising regions that contain ‘better’ shapes. Shape comparison within DMM is done by employing a segmentation complexity measure $\Pi(S, \alpha, D)$ defined by a shape matrix S and a corresponding intensity α and a difference matrix D as follows: apply bDif to segment the residual difference matrix $D - \alpha\,\mathrm{Diff}(S)$ and then define $\Pi(S, \alpha, D) = \sum_{k=1}^{K'} \alpha_k + 7 \times K'$ where K' is the number of shapes in the corresponding segmentation of $D - \alpha\,\mathrm{Diff}(S)$ (7 is a weighting factor that balances the relative importance of beam-on time and segment count). Low values of Π therefore correspond to ‘good’ shapes and a Π value of 0 indicates that a segmentation has been completed by using shape S to produce a 0 residual matrix. In order to modify DMM to take into account TnG effects, Π could be modified to include a penalty term associated with those effects.

DMM uses this segmentation complexity index in the rowwise construction (as described in the flow chart in figure 3) of an $R \times C$ shape table (ST), which accumulates the results of searches for good segmentations and at completion contains the results with the smallest number of shapes in its last row. Since each element in each row corresponds to a shape used in a segmentation, the row parameter R (set to 85 in our runs) simply sets an upper bound on the number of shapes in the segmentation. The column parameter C determines the number of partial segmentations that will be simultaneously developed. (The value of C plays an important role in determining the run time of the procedure, and for a run time of approximately 3 min (on an 800 MHz Linux computer) per beam angle, the following C values were used for 5-, 10- and 100-intensity-level maps: 40, 20, 2. Note that the corresponding run times on faster 3 GHz processors would be about 1 min per beam angle.) The elements of ST consist of shape, intensity and additional information—specifically, tuples given by $\{S, D - \alpha\,\mathrm{Diff}(S), \Pi(S, \alpha, D), \text{previous\_index}\}$, where previous_index is a pointer to the previous ST entry (if any) that gave rise to residual difference matrix D. Thus, following the



***Use C shapes to construct first shape table row*:**

Construct the first row of the Shape Table (ST) using C shape matrices that approximately minimize Π(S, α, D).

Is some Π in current row = 0 ?

Obtain candidate shapes for the next row:

Create a shape and residual pool from the current ST row. From this pool, construct (S, α, D) tuples, and select the C tuples with smallest Π to construct the tuples for row *Row* + 1.
Set *Row* = *Row* + 1.

Select and output segmentation:

Considering elements in the last row with Π =0, determine an entry with minimum beam time. Follow the corresponding backward chain of shapes and intensities to obtain the segmentation.

Figure 3. Main procedures of DMM metaheuristic. Π denotes a segmentation complexity index constructed via the base heuristic bDif.

pointers back-up from a shape at the bottom of the table that completes a segmentation gives the remaining shapes and intensities in the segmentation.

The initial row of ST corresponds to the C multiple starting points and is generated by setting D to the initial difference matrix $\mathrm{Diff}(A)$, and then generating C shape matrices and corresponding α values that approximately minimize the segmentation complexity measure $\Pi(S, \alpha, D)$. Details of the heuristic used in this approximate minimization are given in appendix B. Additional rows of ST are then constructed by considering alternative continuations of the segmentations 'begun' in the preceding rows, using the bDif-based complexity measure $\Pi(S, \alpha, D)$ to rank these alternatives (see appendix B).

Figure 2 illustrates a shape table and a corresponding set of pointers. In this example, segmentations (not shown) using shapes S_1 and S_3, while initially promising, were later discarded after improved segmentations using shapes S_2 and S_4 were generated by considering alternative ‘continuations’ of segmentations containing the latter two shapes. The best (in terms of segment count) segmentation generated thus occurs in one of the four chains of five segments shown in the figure. Each of these segmentations has the smallest number of segments of any of the segmentations generated by the search process, so DMM is completed by selecting a chain with minimum corresponding beam-on time from this set.

2.2. An ‘exact’ branch-and-cut (BC) approach

We compare our metaheuristic with the heuristic used in Corvus and with a MIP (mixed-integer programming) approach based on a model proposed in Langer *et al* (2001) (extending their model via additional constraints, since the MLC constraints of the Elekta MLC are more complex than those considered by Langer *et al*). In the latter, 5-intensity-level inputs with values 0, 20, …, 100 are scaled by dividing by 20 to produce matrices with entries 0-, …, 5- and 10-intensity-level inputs are similarly scaled by dividing by 10. Binary dose variables $d[i, j, t]$ are used to specify if radiation is delivered at bixel i, j at time t.

Langer *et al* suggested a two-phase method in which beam-on time is minimized in the first phase and the number of segments is minimized (subject to a minimum beam-on time constraint) in the second phase. Unfortunately, the additional constraints (as described below) required to model the Elekta MLC constraints make this MIP approach very problematic for intensity maps with ten or more intensity levels. For difficult clinical cases, this approach yields relatively poor results even for 5-intensity-level maps. Since the BC method is seldom able to produce or verify (within the 2 h of computing time that we allowed) an optimal solution for the problem of minimizing the number of shapes (which is more difficult than minimizing beam-on time), it is best regarded in the context of these problems as a heuristic method. The software used to construct and solve the MIP model was the AMPL modelling language combined with the CPLEX branch-and-cut solver (a leading commercial solver), running on 800 MHz Linux machines, and the results of the implementation are presented below.

As proposed by Langer *et al*, we used binary variables $l[i, j, t]$ that are set to 1 to indicate bixels $[i, j]$ covered by a left leaf at time t, binaries $r[i, j, t]$ for the right leaves and $d[i, j, t]$ for uncovered bixels, leading to the constraint

$$l[i, j, t] + r[i, j, t] + d[i, j, t] = 1.$$

Standard leaf position constraints may then be modelled using these variables (see Langer *et al* (2001)). In order to model some special properties of the Elekta MLC, we need to augment Langer’s model to include several additional types of constraints, some of which we describe in detail below in order to provide some insight into modelling issues.

Mono-shape constraints require that each aperture consist of only one ‘connected’ shape matrix which satisfies the leaf collision constraints. In order to model these constraints, binary variables $delivery[i, t]$ are introduced. A variable of this type is 1 if and only if radiation is being delivered in row i at time t. Therefore, the binary variable *delivery* is forced to be 0 if the ith row is not being used at time t. These properties are enforced by the following constraints, where Cols denotes the number of columns in the intensity map:

$$\text{delivery}[i, t] \leqslant \sum_j d[i, j, t] \leqslant \text{delivery}[i, t] \times \text{Cols}.$$

Binary variables *drop* are used (beginning with row 2) to determine if the preceding row in a shape is non-zero and the current row is 0. This will allow the determination of the end row of a mono-shape. The constraints below will force *drop* to 1 in this case:

$$\text{delivery}[i - 1, t] - \text{delivery}[i, t] \leqslant \text{drop}[i, t].$$

0	0	20	0
0	20	20	0
0	20	20	0
0	0	0	0
0	0	0	0
0	0	0	0

0	0	20	0
0	20	20	20
0	20	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Figure 4. Acceptable shape with the straight-edge constraint (left). Unacceptable shape without the straight-edge constraint enforced (right).

Similarly, the binary variable *jump* is used to determine if the previous row is 0 and the current row is non-zero. This will signify the beginning of a mono-shape. The constraints below will force *jump* to 1 in this case:

$$\text{delivery}[i,t] - \text{delivery}[i-1,t] \leqslant \text{jump}[i,t].$$

To ensure that the mono-shape constraint holds, there can only be one row where a mono-shape begins. This leads to the constraint

$$\sum_i \text{jump}[i,t] \leqslant 1.$$

Finally, the following constraints ensure that once the mono-shape ends, all subsequent rows are 0:

$$\text{delivery}[i+1,t] \leqslant 1 - \sum_{I=2,\ldots,i} \text{drop}[I,t].$$

The following 'diagonal' leaf collision constraints are required by the Elekta MLC, but note that the algebraic model must be written so that these leaf constraints will be applied to the leaf positions only in the case of adjacent non-zero rows (in which case the RHS of the constraints will be 1, so that the leaf position variables on the LHS cannot both have value 1):

$$l[i+1,j,t] + r[i,j+1,t] \leqslant 3 - \text{delivery}[i+1,t] - \text{delivery}[i,t],$$
$$l[i-1,j,t] + p[i,j+1,t] \leqslant 3 - \text{delivery}[i-1,t] - \text{delivery}[i,t].$$

Tongue-and-groove constraints as specified in Langer's paper were not used in the results reported below, since CPLEX could not determine integer feasible solutions of these more complex problems in 12 h of computing time.

We also imposed straight-edge constraints required by our machine specifications which ensure that either the right-hand or left-hand side of a shape line up in one column; creating a straight edge on one side of the shape. This is due to the minimum separation constraint (1 cm) for the Elekta MLC. This constraint needs only be enforced when the entire shape is lying in only the upper or lower half of the total number of rows (see figure 4).

2.3. Clinical test cases and MU calculation

We tested our metaheuristic intensity map segmentation algorithm and the BC implementation on three clinical IMRT cases: (i) head and neck, (ii) pancreas and (iii) prostate. All plans were devised so that the dose per fraction was 1.8 Gy, and a 6 MV beam was used in all three cases. The head and neck plan consisted of five beam angles: 55°, 165°, 245°, 290° and 350°. The prescription dose was 45 Gy and the critical structures were the parotid glands and the spinal

Table 2. Summary of dose and dose–volume constraints for the head/neck, pancreas and prostate cases. The same constraints were used in Corvus versions 4.0 and 5.0.

Structure	Limit dose (Gy)	% volume > limit dose	Min dose (Gy)	Max dose (Gy)
		Head/neck		
Parotid (L)	30	20	9	40
Parotid (R)	15	20	1	23
Cord	30	25	2	40
		Pancreas		
Kidney (R)	25	5	15	35
Cord	20	5	5	30
		Prostate		
Bladder	45	15	10	70
Rectum	45	50	15	70

cord. The pancreas case plan consisted of seven beam angles: 0°, 51°, 103°, 154°, 206°, 257° and 308°. The prescription dose was 63 Gy and the critical structures were the right kidney spinal cord. The prostate case plan consisted of six beam angles: 35°, 80°, 135°, 225°, 280° and 325°. The prescription dose was 75.6 Gy and the critical structures were the bladder and rectum. Plans were designed with 5, 10 and 100 intensity levels and in versions 4.0 and 5.0 of the Corvus treatment planning system. The dose and dose–volume constraints for the critical structures in each of the cases are summarized in table 2. All plans were normalized so that at least 97.5% of the CTV received the prescription dose.

Following dose optimization in the Corvus treatment planning system, the idealized intensity maps corresponding to each beam angle were extracted along with the Corvus MLC controller leaf sequence files for the Elekta SL20 accelerator for each plan. The number of segments obtained from the Corvus (versions 4.0 and 5.0), difference-matrix metaheuristic and BC implementation were compared. Given the relative weighting of each segment within the leaf sequence for a single beam, we calculated the relative 'beam-on time' for the Corvus leaf sequence. The absolute MUs were obtained directly from the treatment plan. The idealized intensity maps were then input into our DMM algorithm to generate a new leaf sequence and the leaf segments were output in the format required by the MLC controller file. Using the relative weighting of each segment, we calculated the relative 'beam-on time' for the DMM leaf sequence. To calculate the absolute MUs required with our leaf sequence, we calculated the ratio between the relative 'beam-on time' obtained from the Corvus leaf sequence and that obtained from our leaf sequence for each beam and then multiplied the absolute plan MUs obtained from Corvus by this ratio. An absolute MU comparison is only provided between Corvus 4.0 and the DMM algorithm for purposes of brevity.

We compared the realistic dose maps obtained from both the Corvus leaf sequence and the leaf sequence obtained with our approach. The calculation was performed using a previously described convolution/superposition-based dose calculation algorithm (Naqvi *et al* 2003) on a 30×30 cm^2 water phantom and at a depth of 2 cm. Following a comparison of the dose maps from each beam angle, we performed a 3D dose calculation on patient CT data sets with the same convolution/superposition dose calculation algorithm and using both the Corvus leaf sequence and the metaheuristic leaf sequence in order to investigate if there was any degradation in the dose distribution.

Table 3. Comparison of metaheuristic DMM with Corvus 4.0 commercial treatment planning system and with 30 and 120 min runs of an 'exact' method (BC30 and BC120). Entries give treatment plan totals and improvement % relative to Corvus 4.0. DNR: x/y indicates that the 'exact' method failed to provide a solution for x out of y beam angles.

	Number of segments (improvement %)			Monitor units (improvement %)		
	Prostate	Head and neck	Pancreas	Prostate	Head and neck	Pancreas
5 Intensity levels						
Corv4	192	201	381	1327	1297	2778
DMM	36 (81%)	40 (80%)	116 (70%)	649 (51%)	773 (40%)	2124 (24%)
BC30	49 (74%)	DNR: 2/5	DNR: 7/7	733 (45%)	DNR: 2/5	DNR: 7/7
BC120	45 (77%)	DNR: 1/5	DNR: 5/7	733 (45%)	DNR: 1/5	DNR: 5/7
10 Intensity levels						
Corv4	206	231	486	1500	1342	2621
DMM	64 (69%)	59 (74%)	124 (74%)	914 (39%)	791 (41%)	1794 (15%)
BC30	DNR: 3/6	DNR: 2/5	DNR: 7/7	DNR: 3/6	DNR: 2/5	DNR: 7/7
BC120	DNR: 3/6	DNR: 2/5	DNR: 6/7	DNR: 3/6	DNR: 2/5	DNR: 6/7
100 Intensity levels						
Corv4	268	281	590	1315	1230	2752
DMM	136 (49%)	140 (50%)	337 (43%)	1291 (2%)	838 (32%)	1977 (28%)
BC30	DNR: 6/6	DNR: 5/5	DNR: 7/7	DNR: 6/6	DNR: 5/5	DNR: 7/7
BC120	DNR: 6/6	DNR: 5/5	DNR: 7/7	DNR: 6/6	DNR: 5/5	DNR: 7/7

3. Results and discussion

The results are organized as follows. We first describe the comparison between our metaheuristic leaf-sequencing algorithm and the Corvus results. The number of segments and MUs are compared for the plans obtained from version 4.0 of Corvus, while only the number of segments are compared (for purposes of brevity and relevance) for the plans obtained with version 5.0. Following this comparison we describe the results obtained from the BC implementation.

3.1. Corvus, version 4.0 and DMM comparison

Table 3 shows a comparison of the cumulative number of segments and MUs for the head and neck, pancreas and prostate cases, respectively. The results are shown for the plans with 5, 10 and 100 intensity levels and are denoted by Corv4 for the results obtained from version 4.0 in Corvus, DMM for our metaheuristic algorithm, BC30 and BC120 for the BC implementation, where BCx denotes that x min were allowed for the branch-and-cut approach. The decreases in the total number of segments when our leaf sequence is applied in comparison with the leaf sequence available in Corvus 4.0 were 80%, 74% and 50% for 5, 10 and 100 intensity levels in the head and neck case. The corresponding decreases in the number of MUs were 40%, 41% and 32% as a function of the number of intensity levels. The decreases in the number of segments for the pancreas case were 70%, 74% and 43% for 5, 10 and 100 intensity levels. The MUs were decreased by 24%, 15% and 28%, respectively. For the prostate case, the number of segments were decreased by 81%, 69% and 49% as a function of 5, 10 and 100 intensity levels. The MUs were decreased by 51%, 39% and 2% as a function of intensity levels. The DMM metaheuristic consistently produces high-quality (and sometimes

Table 4. Comparison of the metaheuristic DMM with the Corvus 5.0 commercial treatment planning system and with 30 and 120 min runs of an ‘exact’ method (BC30 and BC120).

	Number of segments (improvement %)		
	Prostate	Head and neck	Pancreas
5 Intensity levels			
Corv5	40	71	135
DMM	26 (35%)	47 (34%)	87 (36%)
BC30	26 (35%)	DNR: 2/5	DNR: 6/7
BC120	26 (35%)	DNR: 1/5	DNR: 5/7
10 Intensity levels			
Corv5	120	103	231
DMM	58 (52%)	60 (42%)	128 (45%)
BC30	DNR: 2/6	DNR: 4/5	DNR: 7/7
BC120	DNR: 1/6	DNR: 2/5	DNR: 7/7
100 Intensity levels			
Corv5	208	201	427
DMM	144 (31%)	144 (28%)	338 (21%)
BC30	DNR: 6/6	DNR: 5/5	DNR: 7/7
BC120	DNR: 6/6	DNR: 5/5	DNR: 7/7

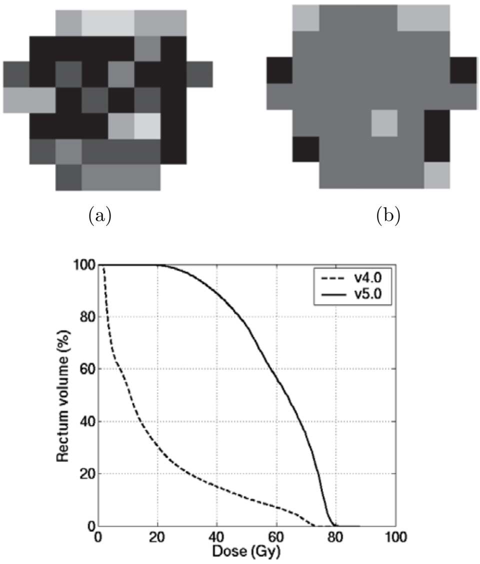
optimal) segmentations within 3 min, and, in all cases, produces a plan with a number of segments that is significantly smaller than the number produced by Corvus 4.0.

3.2. Corvus, version 5.0 and DMM comparison

Table 4 summarizes the comparison between number of segments produced by Corvus 5.0 and our metaheuristic. The reduction in the number of segments obtained with our algorithm in the head and neck case was 34%, 42% and 28% for 5, 10 and 100 intensity levels. In the pancreas case, the decrease in the number of segments with our approach was 36%, 45% and 21% while in the prostate case, the corresponding decrease was 35%, 52% and 31% as a function of intensity levels. In general, a greater reduction was seen in the 10-intensity level cases than with 5 or 100 intensity for version 5.0. The improvement in the number of segments using our approach was less dramatic than with version 4.0. One reason for this performance difference is that some of the *I*-maps generated in version 5.0 were less modulated than those in version 4.0 for the same set of dose and dose–volume constraints. Thus, these *I*-maps are easier to segment than the *I*-maps produced by Corvus 4.0. However, the less modulated *I*-maps result in inferior DVHs for the OARs (as shown in figure 5(c), which compares the Corvus 4.0 and 5.0 dose distributions for the rectum in the prostate case). Note that the volumes of the rectum receiving at least 50 Gy differ by more than 50% in this case. It must be pointed out that these DVHs were extracted directly from the Corvus planning system. They differ from the rectal DVHs shown in figure 11 because they correspond to the non-overlapped volume (with the PTV) of the rectum. The DVHs in figure 11 correspond to the entire volume of the rectum. In either case, the results shown are for consistent volumes in each figure, ensuring a fair comparison.

3.3. Comparison of an ‘exact’ method with DMM

While the branch-and-cut (BC) approach, given a sufficiently large amount of computing time (about 2 h per beam angle), can occasionally produce lower cardinality segmentations than



(a) (b)

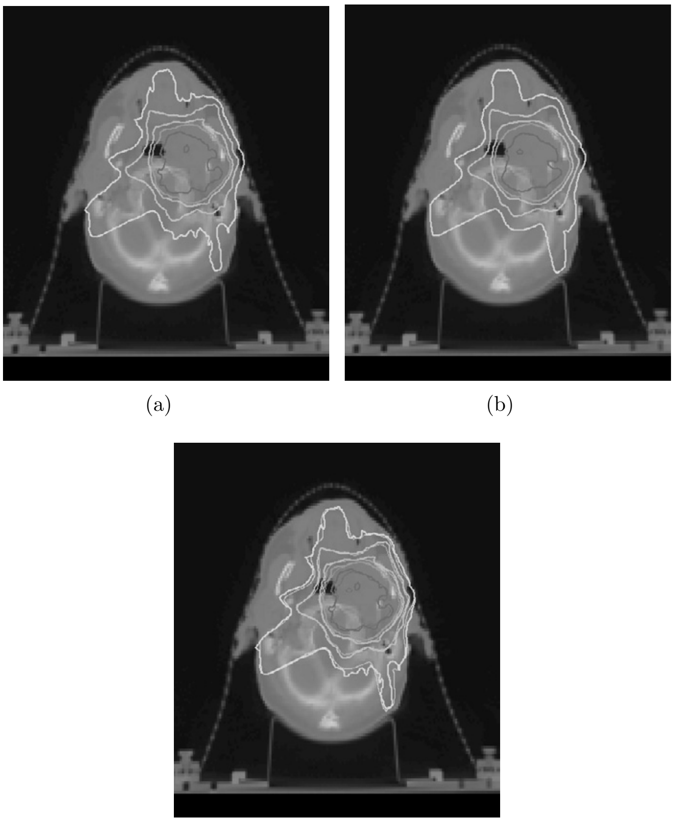
(c)

Figure 5. Comparison of an idealized *I*-map from a sample beam angle for a prostate case from (a) Corvus, version 4.0 and (b) Corvus, version 5.0 illustrating the simpler (less modulated) *I*-maps generated in version 5.0 and (c) the resulting inferior rectum cumulative DVH for the 3D dose distribution.

3 min runs of our metaheuristics for 5-intensity-level maps, BC is 'brittle' in the sense that it sometimes fails to produce any solutions for 5-intensity-level cases (the DNR notation in the tables below stands for 'did not run', indicating that no feasible solution was obtained within the time allowed), and BC generally fails for 10 (or higher)-intensity-level cases. On the other hand, the DMM metaheuristic consistently produces high-quality (and sometimes optimal) segmentations within 3 min, and, in all cases, produces a plan with a number of segments that is smaller than the number produced by Corvus 4.0 and Corvus 5.0. The column headings BC30 and BC120 indicate the branch-and-cut method with 30 min and 120 min time limits. DMM was allowed a time limit of approximately 3 min. Since Corvus does not allow segmentation to be performed as a separate task from the dose optimization in the planning system, it is difficult to determine the CPU time needed to generate the Corvus leaf sequences.

3.4. Calculated dose comparison

We first calculated the dose maps from each beam in a water phantom at a depth of 2 cm. This was done as a qualitative quality assurance check to ensure that the original intensity map was recovered. However, there were two notable differences between the dose maps calculated using the DMM and the Corvus version 4.0 algorithms. First, tongue-and-groove effects were seen in the DMM leaf sequence and minimized in the Corvus leaf sequence. This was because the Corvus leaf sequence forces the leaves to move unidirectionally during step-and-shoot delivery. Second, notable leakage was visible on the Corvus dose maps due to the partial transmission through back-up Y-diaphragm present in the Elekta SL20 linac. Despite these differences in the dose maps the quantity of real consequence was the 3D dose distribution. Figures 6–8 show the comparison between the 3D dose distribution from the



(a) (b)

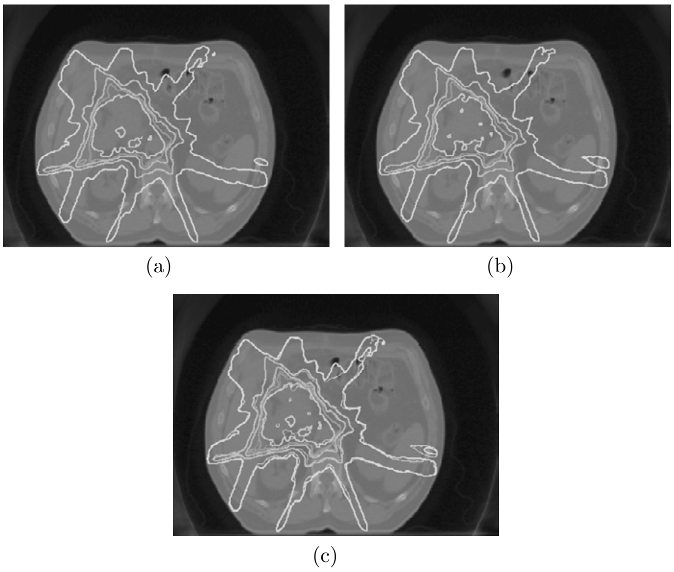
(c)

Figure 6. Head/neck case: 3D dose distribution obtained from (a) original Corvus, version 4.0 and (b) difference-matrix-based leaf sequence. (c) Overlay of dose distributions from Corvus and DMM leaf sequences. The 115%, 100%, 90%, 70% and 50% isodose lines are shown (100% = 45 Gy).

step-and-shoot delivery sequence using the DMM and Corvus version 4.0 algorithms for the head and neck, pancreas and prostate cases, respectively. It was observed that overlays of the dose distributions show that the isodose lines agree very well. Minor discrepancies in the isodose line comparison were not considered clinically significant as evidenced by the DVH plots.

Figures 9–11 show the DVH comparison obtained from the calculated dose distributions using the Corvus 4.0 and DMM leaf sequences. In the head and neck case, the volume of the CTV receiving 45 Gy was 99.8% and 100% for the Corvus 4.0 and DMM leaf sequences, respectively. The volume of the PTV receiving 45 Gy was 93.7% and 92.7% for the Corvus 4.0 and DMM leaf sequences. The volume of the left parotid and cord receiving 30 Gy was 70.2% and 63.3%, and 20.9% and 16.3% with the Corvus 4.0 and DMM leaf sequences, respectively. For the right parotid, the volume receiving 15 Gy was 35.2% and 32.8% for the Corvus 4.0 and DMM sequences.

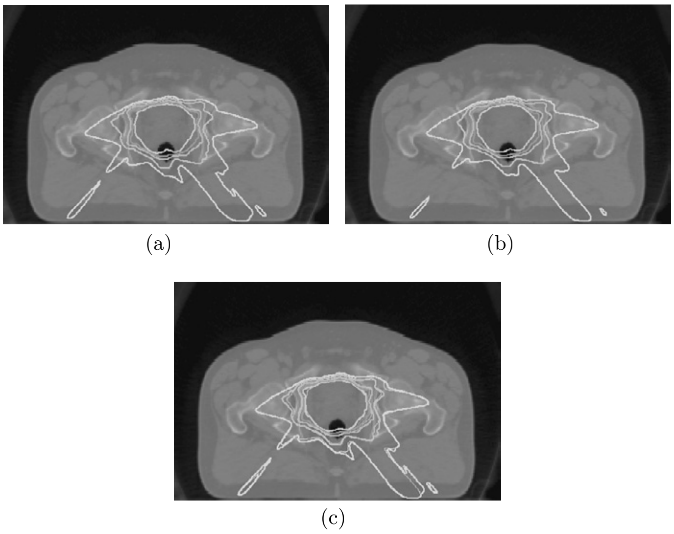
For the pancreas case, the volume of the CTV receiving 63 Gy was 98.8% and 98.4% with the Corvus 4.0 and DMM sequences, respectively, while the corresponding volumes for the PTV were 78.5% and 76.6%, respectively. The volume of the right kidney receiving



(a) (b)

(c)

Figure 7. Pancreas case: 3D dose distribution obtained from (a) original Corvus, version 4.0 and (b) difference-matrix-based leaf sequence. (c) Overlay of dose distributions from Corvus and DMM leaf sequences. The 110%, 100%, 90%, 70% and 50% isodose lines are shown (100% = 63 Gy).



(a) (b)

(c)

Figure 8. Prostate case: 3D dose distribution obtained from (a) original Corvus, version 4.0 and (b) difference-matrix-based leaf sequence. (c) Overlay of dose distributions from Corvus and DMM leaf sequences. The 110%, 100%, 90%, 70% and 50% isodose lines are shown (100% = 75.6 Gy).

25 Gy was 41.7% and 36.4% with the Corvus 4.0 and DMM leaf sequences. The volume of the cord receiving 20 Gy was 66.9% and 65.8%, respectively, with the two sequences.

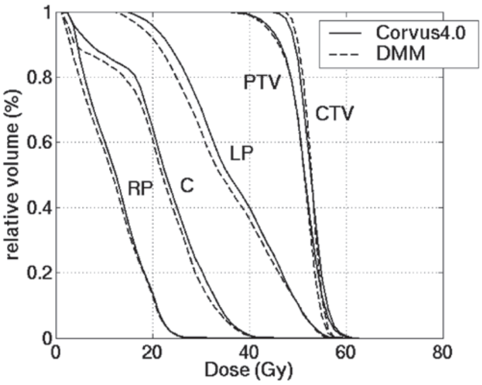


Figure 9. DVH comparison of the Corvus 4.0 and DMM leaf sequences for the head and neck case (CTV: clinical tumour volume, PTV: planning tumour volume, LP: left parotid, C: cord, RP: right parotid).

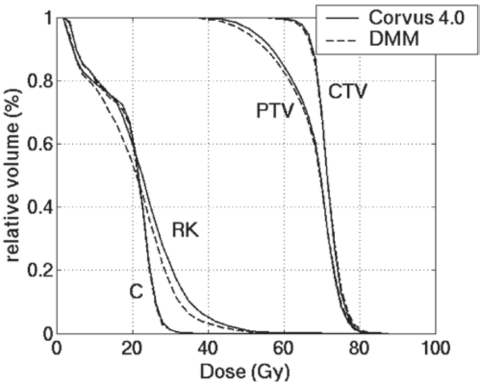


Figure 10. DVH comparison of the Corvus 4.0 and DMM leaf sequences for the pancreas case (RK: right kidney, C: cord).

In the prostate case, the volumes of the CTV and PTV receiving 75.6 Gy were 98.1% and 95.6%, respectively, for both leaf sequences. The volume of the bladder receiving 45 Gy was 56.7% and 53.6% for the Corvus 4.0 and DMM leaf sequences, respectively. The volume of the rectum receiving 45 Gy was 95.7% and 94.9% for the Corvus 4.0 and DMM leaf sequences, respectively. In general, we found that the difference between the DVH values obtained from the Corvus 4.0 and DMM leaf sequences was within 3% for most dose levels.

Figure 12 shows the histogram distribution of the difference dose between the 3D doses obtained from the Corvus leaf sequence and the DMM leaf sequence for the head and neck, pancreas and prostate cases. The standard deviations of the difference dose distributions were 0.46 Gy, 1.05 Gy and 0.51 Gy for the head and neck, pancreas and prostate cases. These may be considered clinically insignificant.

The purpose of this work was to develop a fast heuristic to minimize the number of segments during intensity map segmentation following dose optimization in IMRT. We have also benchmarked our metaheuristic against an integer algorithm, which has been shown to produce superior results to previously published heuristics. This work was carried out

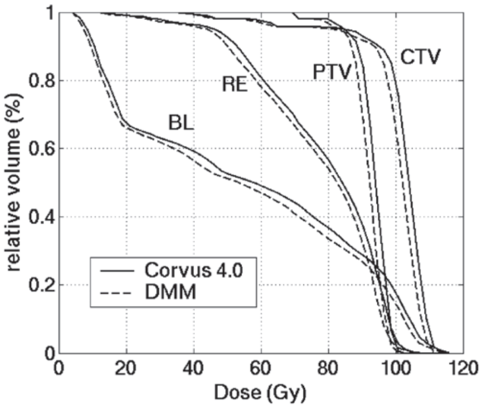


Figure 11. DVH comparison of the Corvus 4.0 and DMM leaf sequences for the prostate case (BL: bladder, RE: rectum).

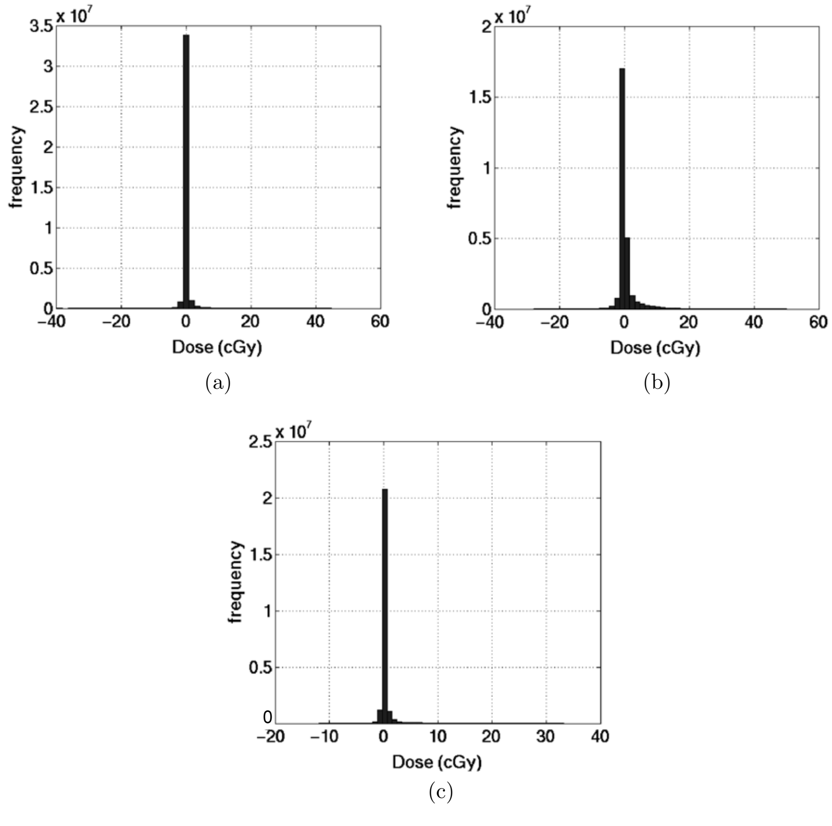


Figure 12. Histogram distribution of the difference in the 3D dose distributions calculated using the Corvus and DMM leaf sequences for the (a) head and neck, (b) pancreas and (c) prostate cases.

by considering the Elekta MLC constraints where the total treatment time is influenced to a greater extent by the number of segments rather than the number of MUs. From an optimization standpoint, it is more difficult to minimize the number of segments than the number of MUs. Monitor unit minimization can be performed by using a linear network flow formulation in which the flow variables correspond to monitor units, whereas segment minimization requires a nonlinear combinatorial approach involving a fixed-charge function that is set to 1 if a segment is used and 0 otherwise.

In this work, we do not consider tongue-and-groove constraints. Intuitively, we believe that implementing such constraints will lead to an increase in the number of segments. Further, even though there are differences in the calculated dose maps between the original dose maps produced from the Corvus planning system, such differences do not translate into significant differences when one considers the 3D dose distribution. Our current research is investigating the use of tongue-and-groove constraints. Implementation of tongue-and-groove constraints can be accomplished via our approach since it generates segments sequentially. In this segment construction process, potential rows for a segment would be compared to rows of previously generated segments and rows violating tongue-and-groove constraints would be penalized.

The computational results presented above demonstrate that for a variety of difficult clinical cases the DMM metaheuristic is able to achieve a dramatic reduction in aperture count relative to the procedures used in the Corvus 4.0 commercial treatment planning software. The number of apertures generated by DMM for the 5-intensity-level intensity maps is generally 70–80% lower than the number generated by Corvus, version 4.0. The decrease in the number of apertures achieved was 70% for the 10-intensity-level map and about 40–50% for the 100-intensity-level maps. For the less differentiated (flatter and hence simpler) intensity maps associated with Corvus 5.0 for the same clinical cases, the DMM aperture count improvements are generally not as dramatic, but are approximately 35%, 40–50% and 20–30% for the 5, 10 and 100 intensity levels. The results obtained for the CPLEX implementation of the branch-and-cut (BC) approach demonstrate that this method is capable of generating results comparable to DMM for the simplest cases (such as some of the 5-intensity-level maps), but BC generally fails to produce any useful results for cases of medium to high difficulty (in terms of either number of intensity levels or variability within the intensity map).

There are two aspects to evaluating an IMRT plan: quality of the dose distribution and efficiency of delivery. The reduction in the number of segments with version 5.0 of Corvus also translates into an inferior plan as shown above. It appears that the Corvus system smoothes the original intensity maps following dose optimization to an extent that it generates simpler (or less modulated) maps. This process can degrade the quality of the treatment plan. The trade-off here is the quality of the dose distribution for a more efficient delivery. We believe that the simplification of the intensity map in order to reduce the number of segments is unnecessary if a more efficient leaf-sequencing or intensity map segmentation algorithm is applied. Our algorithm does this precisely as evidenced by a comparison of the DMM results with the Corvus version 4.0 results. As a result, the smoothing of the original intensity maps need not be so constrained. Our results suggest that the DMM algorithm is capable of delivering a given IMRT plan efficiently (i.e., by reducing the number of segments) without sacrificing the quality of the plan.

4. Conclusion

We have demonstrated that a metaheuristic based on difference matrices can outperform both commercial treatment planning systems and ‘exact’ integer programming approaches to intensity map segmentation. Commercial systems are fast, but often produce segmentations with large numbers of apertures, whereas the branch-and-cut method for integer programming seeks an optimal segmentation, but in many cases had difficulty constructing such a result

A							
0	0	80	100	100	80	40	0
0	80	100	80	60	100	100	40
0	80	60	60	60	80	40	40
0	100	60	60	60	60	100	60
60	60	80	80	80	80	80	0
20	40	20	20	40	80	20	0
0	100	60	80	100	100	100	0
0	40	80	100	80	80	0	0
0	0	60	100	40	0	0	0

$D = Diff(A)$, $Count(D) = 46$								
0	0	80	20	0	-20	-40	-40	0
0	80	20	-20	-20	40	0	-60	-40
0	80	-20	0	0	20	-40	0	-40
0	100	-40	0	0	0	40	-40	-60
60	0	20	0	0	0	0	-80	0
20	20	-20	0	20	40	-60	-20	0
0	100	-40	20	20	0	0	-100	0
0	40	40	20	-20	0	-80	0	0
0	0	60	40	-60	-40	0	0	0

Figure 13. A 5-intensity-level map (A) and its difference matrix ($D = \text{Diff}(A)$).

$40S$							
0	0	0	40	40	40	0	0
0	0	0	0	0	40	40	40
0	0	0	0	40	40	40	40
0	0	0	0	0	0	40	0
0	0	0	0	0	40	40	0
0	0	0	0	0	40	0	0
0	40	40	40	40	40	40	0
0	0	40	40	40	0	0	0
0	0	0	40	40	0	0	0

$40Diff(S)$								
0	0	0	40	0	0	-40	0	0
0	0	0	0	0	40	0	0	-40
0	0	0	0	40	0	0	0	-40
0	0	0	0	0	0	40	-40	0
0	0	0	0	0	40	0	-40	0
0	0	0	0	0	40	-40	0	0
0	40	0	0	0	0	0	-40	0
0	0	40	0	0	-40	0	0	0
0	0	0	40	0	-40	0	0	0

Figure 14. Initial shape matrix S (with beam weight 40) and its difference matrix.

within a clinically acceptable time frame. Our difference-matrix-based heuristics are reliable and fast, and yield segmentations that are comparable to or match the best results generated by 2 h runs of the branch-and-cut approach. We also validated the clinical quality of the segmentations that we obtain by comparisons with the results of commercial treatment planning systems.

Acknowledgments

We would like to acknowledge the financial support from NSF grants DMI-0100220, DMI-0400294 and DMI-0365557.

Appendix A. Example of an idealized intensity map, its difference matrix and the initial step of bDif

Constructing the difference matrix for an I-map: assuming a baseline dose of 0 for bixels outside of the I-map, zero end columns are added to the left (column index = 0) and right (column index = $n + 1$) borders of the original matrix before the difference computation). Algebraically, if A denotes an $m \times n$ matrix, then the difference matrix D_A is the $m \times (n + 1)$ matrix of *column-wise forward differences* with entries $d_{i,j} = a_{i,j} - a_{i,j-1}$, where $a_{i,0} = 0$, $a_{i,n+1} = 0$ and $1 \leqslant j \leqslant n+1$. Note that the difference process is invertible in the sense that the original matrix can be computed from the difference matrix, so no information is lost in the construction of the difference matrix. In the algorithmic discussion we focus on difference-matrix processes, but note that the linearity and invertibility of the Diff operator implies that 'segmentation' of a difference matrix is equivalent to segmentation of the corresponding intensity matrix. Figure 13 shows a 5-intensity-level map and its difference matrix. The Count operator counts the number of non-zeros.

$A - 40S$							
0	0	80	60	60	40	40	0
0	80	100	80	60	60	60	0
0	80	60	60	20	40	0	0
0	100	60	60	60	60	60	60
60	60	80	80	80	40	40	0
20	40	20	20	40	40	20	0
0	60	20	40	60	60	60	0
0	40	40	60	40	80	0	0
0	0	60	60	40	0	0	0

$D' = D - 40Diff(S)$, $Count(D') = 40$								
0	0	80	-20	0	-20	0	-40	0
0	80	20	-20	-20	0	0	-60	0
0	80	-20	0	-40	20	-40	0	0
0	100	-40	0	0	0	0	0	-60
60	0	20	0	0	-40	0	-40	0
20	20	-20	0	20	0	-20	-20	0
0	60	-40	20	20	0	0	-40	0
0	40	0	20	-20	40	-80	0	0
0	0	60	0	-20	-40	0	0	0

Figure 15. Residual matrices after extraction ($A - 40S$ and $D - 40\,\mathrm{Diff}(S)$).

Table 5. Additional properties used in constructing shapes.

M_i = the 1-norm (sum of absolute values) of row i in D
$N = \{i \mid M_i > 0\}$ = the indices of non-zero rows of D
$\mathrm{Max}(D) = \max\{M_i \mid i \in N\}$ = the norm of the largest row in D
$\mathrm{Min}(D) = \min\{M_i \mid i \in N\}$ = the smallest non-zero row norm in D
$\mathrm{Bound}(D) = \max(D) - \min(D)$ = a measure of the variation of the entries of D

In the first step of bDif, a deliverable aperture–intensity pair is generated with a goal of having a small number of non-zero entries in the residual intensity map that remains after the extraction of the weighted aperture from the original intensity map. The next two figures illustrate this step and the resulting residual intensity map.

Appendix B. Constructing aperture–intensity pairs in bDif and DMM

The following additional properties found in table 5 are considered in the procedures used to construct shapes.

To begin the bDif shape construction process with an initial row, we consider each non-zero row i in D and generate one or more row shapes based on the application of heuristics that focus on the maintenance or improvement of the value of the Bound measure. We then construct a pool of two-row shapes by adding second rows that satisfy the aperture constraints. We limit the pool size by keeping at most the P shapes giving the lowest values for $\mathrm{Count}(D - \alpha\,\mathrm{Diff}(S))$. (In our results, the following values of v were used for 5-, 10- and 100-intensity-level maps: 20, 10, 3.) Repeating this local search procedure, we continue to add one row at a time to the current partial shapes. At the end of this process, we select a completed shape from the pool and extract it from the residual intensity matrix using an intensity value as described above. The shape generation and extraction procedure is then repeated until the intensity map has been segmented by bDif. Figure 14 shows a shape matrix S scaled by α and its difference matrix, $\alpha\,\mathrm{Diff}(S)$. Figure 15 shows the resulting matrix $A - \alpha S$ and its difference matrix when we extract αS from the intensity map A shown in figure 13.

In DMM, the ST entries are generated rowwise. For α values, we consider the smallest v distinct elements in absolute value in $\mathrm{Diff}(A)$, where v is an adjustable parameter in the heuristic. (In our results, the following values of v were used for 5-, 10- and 100-intensity-level maps: 2, 4, 15.) Next, a pool of shape matrices is constructed corresponding to these α values using the shape construction method in bDif. The residuals $D - \alpha\,\mathrm{Diff}(S)$ from the current row form the difference pool. (For the construction of first row, the difference pool contains only a single element, $\mathrm{Diff}(A)$.) The (S, α, D) tuples for the next row are generated by forming all possible tuples from the three pools and then selecting from this set the (S, α, D) tuples that yield the best C values of the segmentation complexity function Π. These tuples, augmented by previous_index pointers to appropriate entries in the preceding row, form the tuples of the next row. Once a row of ST has been completed that contains at least one Π value of 0, DMM then terminates by using previous_index values to backtrack (starting at an entry with Π value of 0 that has minimum beam time) through ST to obtain the final segmentation.

References

Boland N, Hamacher HW and Lenzen F 2004 Minimizing beam-on time in cancer radiation treatment using multileaf collimators *Networks* **43** 226–40

Bortfeld T R, Kahler D L, Waldron T J and Boyer A L 1994 X-ray field compensation with multileaf collimators *Int. J. Radiat. Oncol. Biol. Phys.* **28** 723–30

Boyer A L and Yu C Y 1999 Intensity-modulated radiation therapy with dynamic multileaf collimators *Semin. Radiat. Oncol.* **9** 48–59

Engel K 2003 A new algorithm for optimal multileaf collimator field segmentation TR D-18051 (Germany: University of Rostock)

Evans P M, Hansen V N and Swindell W 1997 The optimal intensities for multiple static multileaf collimator field compensation *Med. Phys.* **24** 1147–56

Galvin J M, Chen X-G and Smith R M 1993 Combining multileaf fields to modulate fluence distributions *Int. J. Radiat. Oncol. Biol. Phys.* **27** 607–705

Langer M, Thai V and Papiez L 2001 Improved leaf sequencing reduces segments or monitor units needed to deliver IMRT using multileaf collimators *Med. Phys.* **28** 2450–8

Luan S, Wang C, Hu X S, Naqvi S A, Yu C X and Lee C L 2004 A new MLC segmentation algorithm/software for step-and-shoot IMRT delivery *Med. Phys.* **31** 695–707

Naqvi S A, Earl M A and Shepard D M 2003 Convolution/superposition using the Monte Carlo method *Phys. Med. Biol.* **48** 2101–21

Que W 1999 Comparison of algorithms for multileaf collimator field segmentation *Med. Phys.* **26** 2390–6

Siochi R A 1999 Minimizing static intensity modulation delivery time using an intensity solid paradigm *Int. J. Radiat. Oncol. Biol. Phys.* **43** 671–80

Xia P and Verhey L J 1998 Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments *Med. Phys.* **25** 1424–34