View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Strathclyde Institutional Repository

Strathprints Institutional Repository

Akartunali, Kerem and Mak-Hau, Vicky and Tran, Thu (2015) A unified mixed-integer programming model for simultaneous fluence weight and aperture optimization in VMAT, Tomotherapy, and Cyberknife. Computers & Operations Research, 56. pp. 134-150. ISSN 0305-0548 , http://dx.doi.org/10.1016/j.cor.2014.11.009

This version is available at http://strathprints.strath.ac.uk/50513/

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (http://strathprints.strath.ac.uk/) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to Strathprints administrator: strathprints@strath.ac.uk

A Unified Mixed-Integer Programming Model for Simultaneous Fluence Weight and Aperture Optimization in VMAT, Tomotherapy, and CyberKnife

Kerem Akartunalı
Department of Management Science, University of Strathclyde, Glasgow, G1 1QE, United Kingdom, kerem.akartunali@strath.ac.uk

Vicky Mak-Hau
School of Information Technology, Deakin University, Burwood, Vic 3125, Australia, vicky.mak@deakin.edu.au

Thu Tran
Andrew Love Cancer Centre, Barwon Health, Geelong, Vic 3220, Australia, thut@barwonhealth.org.au

In this paper, we propose and study a unified mixed-integer programming model that simultaneously optimizes fluence weights and multi-leaf collimator (MLC) apertures in the treatment planning optimization of VMAT, Tomotherapy, and CyberKnife. The contribution of our model is threefold: i. Our model optimizes the fluence and MLC apertures simultaneously for a given set of control points. ii. Our model can incorporate all volume limits or dose upper bounds for organs at risk (OAR) and dose lower bound limits for planning target volumes (PTV) as hard constraints, but it can also relax either of these constraint sets in a Lagrangian fashion and keep the other set as hard constraints. iii. For faster solutions, we propose several heuristic methods based on the MIP model, as well as a meta-heuristic approach. The meta-heuristic is very efficient in practice, being able to generate dose- and machinery-feasible solutions for problem instances of clinical scale, e.g., obtaining feasible treatment plans to cases with 180 control points, 6,750 sample voxels and 18,000 beamlets in 470 seconds, or cases with 72 control points, 8,000 sample voxels and 28,800 beamlets in 352 seconds. With discretization and down-sampling of voxels, our method is capable of tackling a treatment field of 8000 $cm^3 \sim 64000$ cm^3, depending on the ratio of critical structure versus unspecified tissues.

Key words: OR in medicine; Integer programming; Heuristics; Radiotherapy treatment planning; Metaheuristics; Lagrangian relaxation.

---

1. Introduction

Radiation therapy or radiotherapy has become one of the most common treatment methods for cancer, besides chemotherapy and surgery, with almost two-thirds of all cancer patients expected to have radiotherapy at some stage in their treatment plan [11]. In this treatment, high-energy radiation is used to shrink tumors and kill cancer cells, where radiation damages the DNA of cancer cells [17]. The radiation can be delivered either by radioactive source(s) placed in or near the tumor, called *brachytherapy*; or by a machine outside the body, called *external radiation therapy*, the most common form of radiotherapy. Since our focus is external radiotherapy, we will refer to that simply as radiotherapy in the remainder of the paper.

Figure 1: Gantry that can rotate by 360 degrees.

In radiotherapy, radiation beams are produced by a Linear Accelerator (LINAC), aimed towards the tumor and its surrounding tissues where cancer may have spread. The LINAC is mounted on a gantry, and the gantry rotates the source of radiation beams around the body of a patient. Volumetric-modulated arc therapy (VMAT), Tomotherapy and CyberKnife are recent major advances in external beam radiotherapy.

In VMAT, the gantry can rotate around a patient's body by 360° in a co-planar manner (see Figure 1). Co-planar treatments are possible through rotation of the LINAC couch. Radiation is continuously delivered through one or multiple arcs (see, e.g., Elekta Infinity [9]

or Varian's RapidArc [32]). An "arc" does not necessarily have to be a full 360° rotation. In Tomotherapy, the source of radiation will continuously rotate around the body of a patient in a helical manner (see [2]), hence it is capable of delivering non-coplanar beams. In CyberKnife, the source of radiation is mounted on a robotic arm, and therefore it can deliver radiation beam from almost any point in space (see [1]). In addition to cancer treatment, VMAT and Tomotherapy have also been used for Total Marrow Irradiation (TMI) in reducing Leukemia relapse ratio (see [10] and [38]).

The modulation of the radiation beams is carried out using collimators. For VMAT, a multi-leaf collimator (MLC) is mounted in the LINAC head. The MLC is made up of leaves, which will block the radiation and are arranged parallel to each other in two sets of opposing banks. These leaves can move independent from each other and can create customised beam shape by positioning themselves in a planned position. The radiation field formed by the MLCs are known as *aperture* (see Figure 2). For example, if an aperture is formed by 20 leaves, each 1cm thick, and we have 20 leaf positions, then the MLC is said to have 400 bixels (or *beamlets*).

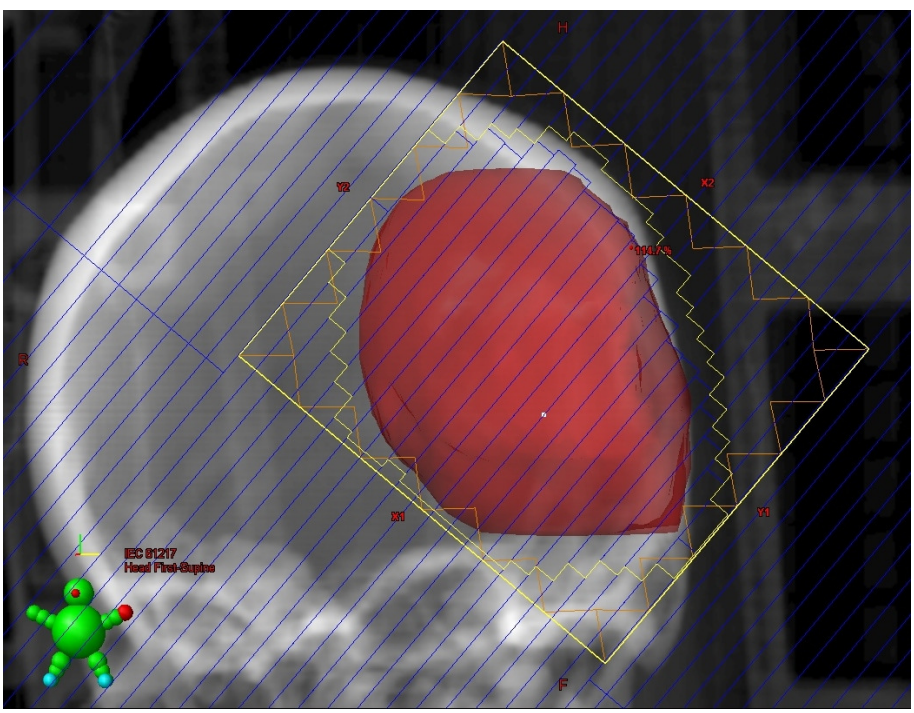

Figure 2: An MLC can form a shape that resembles the shape of the tumor.

On the other hand, TomoTherapy uses simply binary MLCs, i.e., each leaf has only two positions, open or close. The equipment often uses 32 leaves, seen as rows in a beam's

eye view as per Figure 2. CyberKnife has various collimators, e.g., the M6 FI model uses multiple circular collimators of different sizes (called IRIS collimators), whereas the M6 FM model uses an MLC called InCise.

Since MLCs are common in all three machines, we focus our model on optimizing the radiation intensity or fluence weight as well as the MLC aperture simultaneously. Some MLCs do not allow *interdigitation*, that is, the left leaf of a row cannot collide with the right leaf of the neighbouring rows, and vice versa. This is also known as the *interleaf constraints* in the literature. Some MLCs allow full interdigitation, while some allow no interdigitation or no interdigitation with a gap. There are also other machinery constraints for VMAT, such as a limit to the speed a leaf can move in consecutive control points. Such a restriction can be translated to the number of positions, or columns, a leaf is allowed to move between control points before the MLC carriage shifts to enable further proximal/distal positions.

1.1 Recent developments in treatment planning optimization

To plan a radiotherapy treatment that involves the use of an MLC, one must decide from *where* the radiation beams should be delivered, the *intensity* of radiation that should be delivered from each location, and *what apertures* the MLC should form at those locations. The general goals of a treatment is: i. for tumors to receive enough radiation so that they can be eliminated, ii. for organs at risk (OARs) to be spared from radiation as much as possible for minimal damage to healthy tissue, and iii. for the overall treatment time to be as short as possible for patient's comfort. The tumor area is usually given a margin that will cover some surrounding tissues where cancer may have spread to and to compensate for motion and setup. This area is called the Planning Target Volume (PTV). Before the treatment planning optimization takes place, a radiation oncologist prescribes the dose limits to the different structures: a lower bound on the dose to the PTVs (or a volume constraint such as "at least 95% of the PTV must receive a dose of 73.7 Gray (Gy) or above"); and an upper bound on the OARs (or volume limits such as "no more than 35% of the bladder can receive more than 40 Gy of radiation").

The question of *where* will be answered by a set of locations, commonly referred to as the *control points* (CPs). The question of *intensity* at CP k is determined by the dose rate (r_k, in Monitor Unit (MU) per second), and the gantry speed (s_k, also in MUsec). In recent literature, e.g., [25] and [29, 30], a value called *fluence weight* that equals to r_k/s_k is used for simplicity, and we also use this concept in this paper for consistency. Finally the answer to

what aperture the MLC should form is where combinatorial optimization comes in. Further, there is a limit in the maximum fluence weight that can be delivered from each control point. For VMAT, the change in fluence weight is also limited in consecutive control points.

Notice that for all three treatment modalities, only one MLC aperture is considered at each control point. On the other hand, in intensity-modulated radiotherapy (IMRT), a widely used form of radiotherapy with machinery similar to VMAT, there are very few control points (usually only 5-7) in each treatment, but multiple MLC apertures are used at each control point.

Traditionally, fluence weights and MLC apertures are optimized separately. In IMRT, three separate optimization problems are solved (see, e.g., [13], [8], and [7]). First, a *beam-angle optimization problem* is decided. These angles are either predetermined by an experienced treatment planner or calculated by solving some optimization problems (see, e.g., [22, 35, 39]). The output is a number of angles (usually 5-7, at most around 10 for the most complex cancer cases, i.e., head and neck) from where radiation is delivered. After that, a *fluence map optimization* problem is solved [26], which will provide us with one intensity matrix for each beam-angle (control point). Finally, a *realization problem* is solved to find the right MLC apertures that will decompose the intensity matrices, and to find the most time-efficient way to do so ([18]). The field of minimizing treatment times for the step-and-shoot MLC radiotherapy practically began with the work of [36]. The most recent advances in the minimizing of total treatment time can be found in [5, 18, 31].

Since the underlying mathematical problem for the VMAT is different, methods developed for the IMRT cannot be directly applied to the VMAT. The difference mainly stems from the interconnectivity between different control points and the computational complexity due to higher degree of freedom in VMAT, as noted by [37]. To our knowledge, the treatment planning optimization of the VMAT has rarely been studied in the field of mathematical programming/operations research, although the problem offers interesting challenges. The most recent treatment planning optimization algorithms can be found in [25] (a greedy heuristic) and [29, 30] (a hybrid nested partitioning heuristic). On the other hand, recent years have seen a growing interest from medical physicists: [6, 24, 20] propose various heuristics for handling dose limits as soft constraints, whereas [34] propose a two-stage algorithm using shortest paths and [23] extend a one-stage algorithm originally proposed for IMRT problems. We refer the interested reader to [37] for an excellent review discussing detailed clinical aspects of and different solution methodologies proposed for arc therapy, as well as

an invaluable discussion comparing different therapies for different cancer cases.

In [25], a mixed-integer nonlinear programming model is presented. The method iterates between two optimization problems: First, given a set of control points and their respective MLC apertures, the fluence weights for all control points are optimized. Then, a column generation approach is used to find the most promising control point and its associated MLC aperture. The objective function used therein is to minimize the deviation of the calculated dose from the prescribed dose. Hence, if the objective value of a solution is not zero, then it means that there exist violations in dose limits. A down-sampled voxel grid is implemented– one in every two voxels is sampled along each of the three dimensions in critical structures and one in every four in unspecified tissues. The voxels are of size $4 \times 4 \times 2.5\ mm^3$. Real clinical cases were tested using GPU computing and solution times are shown to be very fast. Computational results indicated that the dose constraints for most structures could be reasonably satisfied except for the case with the bladder.

The methods of [29, 30], on the other hand, present a mixed-integer programming model similar to the model presented in this paper. The variable sets are adopted from [16] for the IMRT, allowing interdigitation and only enforcing the dose lower bound for PTV as hard constraints. When solving the MIP problem, however, these constraints are relaxed in a Lagrangian fashion in the solution methodology, and a generalized Benders decomposition and hybrid nested partition method is used as the solution methodology. The authors present the consecutive control point leaf movement constraint in their model, however, for a problem instance we have tested, there seems to be violations of these constraints present. Their algorithm is tested on 10 randomly generated non-clinical instances.

In CyberKnife, [27] presents a multi-criteria optimization approach to optimize the fluence weights, solving linear programming problems in each step of the multi-criteria optimization procedure, with dose constraints dualized in the objective function and solved by a one-step Lagrangian relaxation with pre-determined dual multipliers.

In TomoTherapy, while there are publications in the optimization of treatment planning parameters such as field width, pitch factor, and modulation factor (see, e.g., [28] and [12]), to the best of our knowledge, there is no integer programming/combinatorial optimization methods specifically designed for TomoTherapy treatment planning. It is, however, possible to apply a VMAT treatment planning optimization method to helical tomotherapy. There will be many more control points to consider, however, the MLC apertures are much easier, as each leaf has only two possible positions, open and close.

1.2 Contributions of our paper

Our paper makes the following contributions to the radiotherapy treatment optimization literature.

1. Both our MIP model and the meta-heuristic method optimizes the fluent weights and the MLC apertures simultaneously. Most methods in the present literature either optimize only the fluence weights (using beam-eye-view as MLC apertures) or optimize the two iteratively. Furthermore, our models involve the no-interdigitation (interleaf) constraints, which can simply be removed if the MLC considered allows interdigitation. With a simple modification of the interleaf constraints, our model can also accommodate the case of no interdigitation with a gap.

2. Our MIP model can be used as a quality analysis tool for existing heuristic algorithms, as well as to confirm infeasibility of a specific patient's case. Most of the existing algorithms dualize the dose limits, reducing the problem to minimizing deviation from prescribed dose limits. With a heuristic solution, even though computation times are typically short (e.g., [25]), when the objective value does not return a zero, one can never know whether the dose prescription is simply infeasible, or that the heuristic method failed to find a solution that is dose-feasible. Our approach, on the other hand, indicates clearly when a given dose prescription is simply infeasible.

3. Although our MIP model can currently solve problems of moderate size, advances in optimization software, computer hardware, and parallel computing are taking place rapidly, increasing the size of problems that can be tackled every day. We also investigate customized exact methods with better computational performance for our MIP model in a companion paper [4]. Moreover, treatment planning often happens 2-3 weeks before the actual treatment is carried out, allowing time to extensively search for a solution that is dose-feasible, rather than to provide a quick solution in a few seconds that does not necessarily comply all dose requirements. As noted by Dr. Robert D. Timmerman, vice chair of radiation oncology at the University of Texas Southwestern Medical Center, radiotherapy plans will be significantly improved in the future, "simply because current outcomes are unsatisfactory to patients" [11]. Therefore our primary focus in this paper is on ensuring dose-feasible solutions whenever possible.

4. For faster solutions, we also provide several heuristic methods based on the MIP model, as well as a meta-heuristic approach. With the meta-heuristic, we were able to solve problems with up to 180 control points for a treatment field with 6,750 sample voxels and 18,000 beamlets; and, in another instance, 72 control points with 8,000 voxels and 28,800 beamlets, for which the first dose- and machinery-feasible solutions were returned after 470 seconds and 352 seconds, respectively. Note that the sizes of problems we have tackled are comparable to those found in the literature, see, e.g., [25]. Using the same discretization and down-sampling of voxels used by [25], our method is capable of optimizing a treatment field of 8000 $cm^3 \sim 64000$ cm^3, depending on the ratio of critical structure versus unspecified tissues. Finally, any solution returned by the heuristic will have <u>neither</u> violations to the prescribed dose <u>nor</u> violations to the machinery restrictions.

1.3 Organization of the Paper

In Section 2, we will present an integer programming formulation of the problem, which initially has non-linear terms but can be linearized with additional variables and constraints. We will also discuss some valid inequalities that strengthen this formulation. In Section 3, we will discuss the polyhedral properties of some key subproblems, and also the strength of the inequalities presented in the previous section. We will present all proposed solution methodologies in detail in Section 4, where we will also briefly discuss possible Lagrangian relaxations of the problem. The empirical strength of these solution methods will be investigated extensively in Section 5. Finally, we will summarize our conclusions and address potential future research areas in Section 6.

2. A Mixed Integer Programming Formulation

In this section, we present a mixed integer programming formulation for the treatment planning optimization of the VMAT.

2.1 Notation and Problem Description

In what follows, we consider a set of given control points. In the case of VMAT, one can consider equally spaced partitions of the 360° coplanar space. For example, a 180 control point problem instance will represent a 2° partition. In the case of TomoTherapy, the control

points can also be taken as equal partitions from the helical rotation. Even though there will be more control points than VMAT if we are to consider fine partitions, there will be significantly less leaf position combinations to consider. In the case of CyberKnife, one can consider a given set of "promising" control points. In theory, one can have a large number of such control points, but impose a limit in the number of control points with a non-zero fluence weight. In this paper, we will consider only the case when a set of control points is given. We now define some notation. Let:

- $I = \{1, \ldots, m\}$ be the index set of the MLC rows;
- $J = \{1, \ldots, n\}$ be the index set of the MLC columns;
- $K = \{1, \ldots, \mathfrak{K}\}$ be the index set of control points;
- $I \times J$ be set of *beamlets* (or *bixels*), each cell (i, j) being a beamlet/bixel;
- $J' = \{0, n+1\} \cup J$, with 0 and $n+1$ being the home positions of the MLC left and right-leaves, respectively;
- δ be the maximum number of columns a MLC leaf is allowed to move in consecutive control points;
- Δ be the maximum amount of change in fluence weights that is allowed in consecutive control points;
- V be the index set of all voxels;
- V_t be the set of voxels in the *target* volumes, i.e., tumor volumes;
- V_o be the set of voxels in the *organs* at risk (one can even define a set each for the critical structures, e.g., V_B for the set of voxels in the bladder(s));
- $\mathcal{L} = \{(\ell, r) \mid \ell, r \in J',\ \ell < r\}$ be the set of feasible left- and right-leaf pairs;
- D^k_{ijv} be the beamlet-based dose deposition coefficient, i.e., the dose, at unit fluence weight, received by voxel v from beamlet (i, j) in the interval defined by control point k (this value can be pre-calculated in the manner as described in [25]);
- L_v be the prescription dose for tumor voxel $v \in V_t$;

- $\bar{d} > L_v$, for $v \in V_t$ be a *desired* dose for voxels in a PTV (e.g., we may have 73.7 Gy as L_v and 79.2 as $\bar{d}$);

- U_v be the maximum dose allowed for any given voxel $v \in V$;

- $y^k_{i(\ell,r)} \in \{0,1\}$ be a decision variable with $y^k_{i(\ell,r)} = 1$ representing the bixels between, but not including, columns ℓ and r in row i in control point k are open;

- $x_v \in \{0,1\}$ be a decision variable with $x_v = 1$ if voxel $v \in V_t$ receives a desired dose of $\bar{d}$ or above, and $x_v = 0$ otherwise;

- d_v be the dose that voxel v receives;

- z^k be a continuous decision variable representing the fluence weight for control point k; and

- $\bar{M}$ be the maximum fluence weight that can be delivered from any control point.

2.2 The Formulation

There are some machinery constraints that are common among VMAT, TomoTherapy, and CyberKnife, and some constraints that vary among the three treatment modalities. For VMAT and CyberKnife, some MLCs allow interdigitation, whereas others do not. With TomoTherapy, since there are only two leaf positions for each leaf, interdigitation is not an issue. With VMAT, there will be a maximum travel distance for the leaves in consecutive control points. Moreover, a maximum fluence weight change is also limited. There are also dose lower limit and upper (or volume) limit constraints. Hence, we propose the following formulation that will capture most of the machinery constraints. We emphasize here that a feasible solution to our model is dose- and machinery-feasible, and is therefore an overall feasible treatment plan.

Formulation 2.1

$$\max\ f(x,y,z) \tag{1}$$

$$\text{s.t.} \sum_{(\ell,r)\in\mathcal{L}} y^k_{i,(\ell,r)} = 1 \qquad \forall i \in I,\ \forall k \in K \tag{2}$$

$$\sum_{\tilde{r}=\ell+1}^{n+1} y^k_{i(\ell,\tilde{r})} + \sum_{\tilde{r}=1}^{\ell}\sum_{\tilde{\ell}=0}^{\tilde{r}-1} y^k_{(i+1)(\tilde{\ell},\tilde{r})} \le 1$$
$$\forall i \in I\backslash\{m\}, \forall \ell \in J'\backslash\{n+1\}, \forall k \in K \tag{3}$$

$$\sum_{\tilde{\ell}=0}^{r-1} y^k_{i(\tilde{\ell},r)} + \sum_{\tilde{\ell}=r}^{n}\sum_{\tilde{r}=\tilde{\ell}+1}^{n+1} y^k_{(i+1)(\tilde{\ell},\tilde{r})} \le 1$$
$$\forall i \in I\backslash\{m\}, \forall r \in J'\backslash\{0\}, \forall k \in K \tag{4}$$

$$\sum_{\tilde{r}=0}^{r-\delta-1}\sum_{\tilde{\ell}=0}^{r-1} y^{k+1}_{i(\tilde{\ell},\tilde{r})} + \sum_{\tilde{r}=r+\delta+1}^{n+1}\sum_{\tilde{\ell}=0}^{\tilde{r}-1} y^{k+1}_{i(\tilde{\ell},\tilde{r})} \le 1 - y^k_{i(\ell,r)}$$
$$\forall i \in I,\ \forall (\ell,r) \in \mathcal{L},\ \forall k \in K \tag{5}$$

$$\sum_{\tilde{\ell}=0}^{\ell-\delta-1}\sum_{\tilde{r}=\tilde{\ell}+1}^{n+1} y^{k+1}_{i(\tilde{\ell},\tilde{r})} + \sum_{\substack{\tilde{\ell}=\ell\\+\delta+1}}^{n}\sum_{\substack{\tilde{r}=\\\tilde{\ell}+1}}^{n+1} y^{k+1}_{i(\tilde{\ell},\tilde{r})} \le 1 - y^k_{i(\ell,r)}$$
$$\forall i \in I,\ \forall (\ell,r) \in \mathcal{L},\ \forall k \in K \tag{6}$$

$$z^k - z^{k+1} \le \Delta \qquad \forall k \in K \tag{7}$$

$$z^{k+1} - z^k \le \Delta \qquad \forall k \in K \tag{8}$$

$$d_v \ge L_v \qquad v \in V_t \tag{9}$$

$$d_v \le U_v \qquad v \in V \tag{10}$$

$$d_v \ge \bar{d}x_v \qquad v \in V_t \tag{11}$$

$$x \in \{0,1\}^{|V_t|}; y \in \{0,1\}^{|I|\times|\mathcal{L}|}; 0 \le z^k \le \bar{M} \tag{12}$$

We first discuss three possible objective functions. The first objective function is suitable for cases where the dose prescription is feasible to be carried out–cases that a solution that satisfy all lower and upper dose limits dose indeed exist. In real-life cases, this is of course not known in advance, in which case the model can be used to prove that no feasible solution exists.

1. With all dose restrictions as hard constraints, we first consider $f(x,y,z) = \sum_{v\in V_t} x_v$, i.e., we maximize the number of voxels in the PTV that receive a desired radiation of $\bar{d} > L_v$, with the help of constraint (52). By maximizing this counter, we are able to

provide an equal weight for each voxel in the target area to receive a good dose, which will favor dose homogeneity as advised in [15].

We also note that the x_v variables can be used to model volume requirements. For example, a requirement such as "at least 95% of PTV must receive a minimum dose of 79.2 Gy" can be written as:

$$\begin{array}{rcl} d_v & \geq & 79.2x_v, \ \forall v \in V_t \\ \sum_{v \in V_t} x_v & \geq & 0.95|V_t|. \end{array}$$

On the other hand, a volume constraint such as "no more than 17% of the bladder can receive 65 Gy or above" can be achieved in the following way. Let V_B be the set of voxels in a bladder, M a large number, we replace the objective function by $\min_{v \in V_B} x_v$ and include the following constraints:

$$\begin{array}{rcl} d_v - 65 & \leq & Mx_v, \ \forall v \in V_B \\ \sum_{v \in V_B} x_v & \leq & 0.17|V_B|. \end{array}$$

In this case, the x_v will be a penalty counter of the number of bladder voxels being delivered a dose over 65 Gy. The next two objective functions are suitable for difficult cancer sites where prescribed doses are known to be typically very hard to be satisfied.

2. Keeping only the PTV dose lower limit constraints (50) as hard constraint, we can minimize the dose delivered to the OARs in the objective function, i.e., $\min \sum_{v \in V_o} d_v$. Alternatively, we can minimize the dose upper limit violations for voxels in OARs, either $\min \{\max_{v \in V_o} \{(d_v - U_v)^+\}\}$ (highest violations) or $\min \sum_{v \in V_o} (d_v - U_v)^+$ (total violation). Note that a simple dose lower limit feasibility check can be performed prior to solving the optimization problem, where if there exists any $v \in V_t$ such that $L_v > \bar{M} \sum_k \sum_{i \in I} \sum_{j \in J} D^k_{ijv}$, then the dose prescription does not have a feasible solution.

3. Keeping only the OAR dose upper limit constraints (51) as hard constraint, we can maximize the dose delivered to the PTV in the objective function, i.e. $\max \sum_{v \in V_t} d_v$. Alternatively, we can minimize the dose lower limit violations for voxels in PTVs, i.e. $\min \{\max_{v \in V_t} \{(L_v - d_v)^+\}\}$ (highest violations) or $\min \sum_{v \in V_t} \{(L_v - d_v)^+\}$ (total violation).

We now explain the rest of the constraints. Constraint (2) ensures that only one shape is used for each control point. This applies to all three treatment methods. Constraints (3) and (4) are the inter-leaf (no interdigitation) constraints that ensure the right (left) leaf in

Row $i+1$ cannot overlap with the left (right) leaf in Row i. Here we note that although some MLCs allow interdigitation (i.e., no interleaf constraints needed), the interleaf constraints in general make the problem computationally harder and they cover a more general set of problems; hence we include them in the formulation. We can easily remove them for TomoTherapy and for MLCs that do allow interdigitation.

Constraints (5) and (6) require that MLC leaves do not move faster than δ columns between successive control points, which applies to VMAT only. Similarly, for VMAT (and potentially for TomoTherapy too), we have the constraints (48) and (49) that limit the difference between fluence weights for consecutive control points to be at most Δ.

Constraints (50) and (51) ensure that the dose lower and upper bounds be satisfied, and (52) determines whether the binary variable x_v can take a value of 1, i.e., the dose applied on voxel v is at least the desired minimum of $\bar{d}$. Finally, (53) indicates bounds and integrality requirements of the variables.

The dose that a voxel receives, denoted by d_v for each $v \in V$ and calculated as in (13), depends on both the MU delivered and the shape of the MLC at each control point:

$$d_v = \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \left(z^k \times D^k_{ijv} \times \sum_{\substack{(\ell,r) \in \mathcal{L} \\ \ell < j < r}} y^k_{i(\ell,r)} \right) \tag{13}$$

In this form, the dose expression is nonlinear and not convex; therefore it would complicate the problem significantly. Although some MINLP techniques could be used, it would be more beneficial if an efficient linearization can be established. We first note that all the nonlinear terms above are bilinear. Moreover, the $z^k \times D^k_{ijv} \times \sum_{(\ell,r) \in \mathcal{L}, \ell < j < r} y^k_{i(\ell,r)}$ terms are special that exactly one of the components is a binary variable and the other a continuous variable. Therefore, we can define a new variable $\bar{z}^k_{ij}$ to indicate the MU amount for control point k and beamlet (i, j) of the MLC, and hence redefine d_v for each $v \in V$ linearly as

$$d_v = \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \bar{z}^k_{ij} \times D^k_{ijv} \tag{14}$$

Recall that $\bar{M}$ is the maximum fluent weight that can be delivered from each control point. To finalize the linearization, we add to the formulation the following four constraints for each

(k, i, j) combination, that is, for all $k \in K, i \in I, j \in J$:

$$\bar{z}_{ij}^k \leq \bar{M} \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell<j<r}} y_{i(\ell,r)}^k \tag{15}$$

$$\bar{z}_{ij}^k \leq z^k \tag{16}$$

$$\bar{z}_{ij}^k \geq \bar{M} \left(-1 + \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell<j<r}} y_{i(\ell,r)}^k \right) + z^k \tag{17}$$

$$\bar{z}_{ij}^k \geq 0 \tag{18}$$

Since $\sum_{(\ell,r)\in\mathcal{L},\ell<j<r} y_{i(\ell,r)}^k \in \{0,1\}$, the variable $\bar{z}_{ij}^k$ will be forced to zero when $\sum_{(\ell,r)\in\mathcal{L},\ell<j<r} y_{i(\ell,r)}^k = 0$, and to z^k when $\sum_{(\ell,r)\in\mathcal{L},\ell<j<r} y_{i(\ell,r)}^k = 1$. Note that this linearization corresponds to McCormick's envelope [19] and is known to give the convex envelope for general bilinear terms. Substituting the linear dose definition (54) into the constraints (50)-(52) and adding the above constraints, the problem is defined as: $z_{VMAT} = \max\{(1)|(x, y, z, \bar{z}) \in X_{VMAT}\}$, where $X_{VMAT} = \{(2) - (53), (55) - (58)\}$.

Given the variety of objective functions and constraints one can study, in this paper, we have chosen to study the combinatorial optimization problem precisely as stated in Formulation 2.1, where the first objective function is used, i.e., $\max \sum_{v\in V_t} x_v$. We also note that we have developed and performed preliminary test on a formulation adopted from the model of [16] for IMRT, wherein for each control point and each row, three sets of variables, $(t_{ij}^k, \ell_{ij}^k, r_{ij}^k \in \{0,1\}^n)$, are used to represent the aperture, with $t_{ij}^k = 1$ if bixel (i, j) of control point k is open, $\ell_{ij}^k = 1$ if bixel (i, j) is occupied by the left leaf, and $r_{ij}^k = 1$ if bixel (i, j) is occupied by the right leaf. (In our model, we need $(n + 2)(n + 1)$ y-variables to do the same job). The leaf logic constraints and the interleaf constraints are presented in [16]. We derived our own consecutive control point leaf movement constraints. From our preliminary experiments on the MIP-based heuristics, however, the performance of the Langer-type model is faster in some cases, but slower in others. It finds better objective values ($\sum_v x_v$) in some cases, but worse objective values in others. A table will be presented in the numerical results section with further discussion. Based on these preliminary results, we decide to use our own formulation presented in this section for further analysis and experiments. The full alternative formulation is presented in detail in the appendices for the sake of completeness.

2.3 Improving the Formulation: Valid Inequalities

In this subsection, we improve the above formulation by considering stronger inequalities that would replace some of the original constraints, and present numerical results to show the strengths of these inequalities. The number of tightened inequalities is equal to the number of original constraints and therefore we will benefit from these inequalities without any obvious additional computational cost. Moreover, as discussed in the upcoming sections, these inequalities are facet-defining for some subproblems while providing computational improvements.

Proposition 2.1 *The following inequality is valid and dominates (52).*

$$d_v - L_v \geq (\bar{d} - L_v)x_v \tag{19}$$

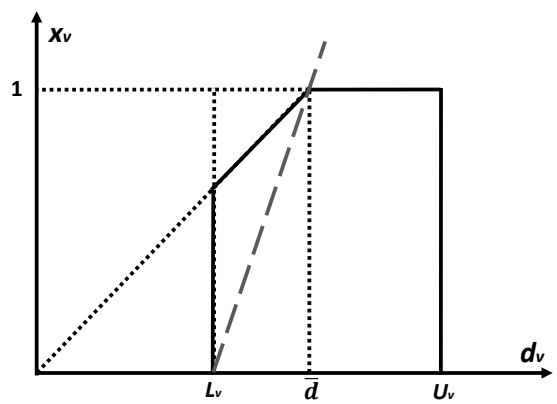

Figure 3: Geometric view of (19) (dashed line) vs. (52) (dotted line starting from the origin)

Note that if $L_v = 0$ holds for any $v \in V$, (19) becomes equivalent to (52). The effect of these strengthened inequalities can be seen in Figure 3 with respect to original constraints (52), when $L_v > 0$. The proof of this proposition is straightforward and is hence omitted here. The following notation is used frequently in the rest of this section.

Definition 2.1 *The cumulative demand from column ℓ to column r is given by: $D^k_{i,\ell,r,v} = \sum_{j=\ell}^{r} D^k_{ijv}$.*

Proposition 2.2 *For all $k \in K,\ i \in I, j \in J$, the inequality*

$$z^k - \bar{z}^k_{ij} \leq \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell\geq j \bigvee r\leq j}} \min\{\bar{M}, \min_{v\in V} \frac{U_v}{D^k_{i,\ell+1,r-1,v}}\} y^k_{i(\ell,r)} \tag{20}$$

is valid for the VMAT problem and dominates (57).

Proof. First of all, note that due to constraint (2), we have:

$$\sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell\geq j \bigvee r\leq j}} y^k_{i(\ell,r)} = 1 - \sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} y^k_{i(\ell,r)}, \quad \forall k\in K,\ i\in I, j\in J.$$

If, for any given $(\ell,r)\in\mathcal{L}$ such that $\ell\geq j$ or $r\leq j$, $y^k_{i(\ell,r)}=1$, then $\bar{z}^k_{ij}=0$, and $z^k\leq \frac{U_v}{D^k_{i,\ell+1,r-1,v}}$ has to be satisfied due to (51). On the other hand, if for any given $(\ell,r)\in\mathcal{L}$ such that $\ell<j<r$, $y^k_{i(\ell,r)}=1$, then $\bar{z}^k_{ij}=z^k$. Therefore, (20) is valid. As $\min\{\bar{M}, \frac{U_v}{D^k_{i,\ell+1,r-1,v}}\}\leq \bar{M}$, (20) dominates (57). □

Corollary 2.1 *For all $k\in K,\ i\in I, j\in J$, the inequality*

$$\bar{z}^k_{ij} \leq \sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} \min\{\bar{M}, \min_{v\in V} \frac{U_v}{D^k_{i,\ell+1,r-1,v}}\} y^k_{i(\ell,r)} \tag{21}$$

is valid for the VMAT problem and dominates (55).

The corollary can be proven similar to Proposition 2.2 and hence we omit the proof.

Proposition 2.3 *The following are valid inequalities for the VMAT problem:*

(a) For $i\in I,\ j,j'\in J,\ k\in K$:

$$\bar{z}^k_{ij} - \bar{z}^k_{ij'} \leq \sum_{\substack{(\ell,r)\in\mathcal{L} \text{ s.t. } (\ell<j<r)\\ \bigwedge(j'\geq r \bigvee j'\leq \ell)}} \min\{\bar{M}, \min_{v\in V} \frac{U_v}{D^k_{i,\ell+1,r-1,v}}\} y^k_{i(\ell,r)} \tag{22}$$

(b) For $i,i'\in I,\ j,j'\in J,\ k\in K$:

$$\bar{z}^k_{ij} - \bar{z}^k_{i'j'} \geq \left(\sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} \min\{\bar{M}, \min_{v\in V} \frac{U_v}{D^k_{i,\ell+1,r-1,v}}\} y^k_{i(\ell,r)} \right) - \bar{M} \tag{23}$$

The proof of the second inequality is similar to the proof presented on p.82 of [33] and hence omitted here. The proof of the first inequality is straight forward due to the fact that $\bar{z}^k_{ij}\leq \min\{\bar{M}, \min_{v\in V}\frac{U_v}{D^k_{i,\ell+1,r-1,v}}\}$ and hence omitted here. Note that the first set of inequalities are facet-defining in some of the subproblems we will discuss in the next section.

We conclude this section with a remark: we run preliminary computational tests to see if and how the inequalities discussed in this section would make a difference. We tested the strengths of the inequalities (19)- (23) individually added to the LP relaxation of the original formulation, as well as the effect of adding all these inequalities together. These results indicated that constraints (19) provide by far the most significant improvement to the LP relaxation bound. The improvements made by (21) and (22) were observed to be mild, and ones made by constraint (20) and (23) were very insignificant. Therefore, we simply replaced (52) with (19) and will use this formulation in the remainder of the paper for all the methods that are based on MIP.

3. Assumptions and Polyhedral Analysis

In this section, we study the polyhedral structure of the MIP. First, we list our assumptions for the model that ensure the model to be both realistic and also non-trivial for the studied cases:

- $\sum_{k\in K, i\in I, j\in J} \bar{M} D^k_{ijv} \geq U_v$, $\forall v \in V$. Otherwise, we can set $U_v = \sum_{k\in K, i\in I, j\in J} \bar{M} D^k_{ijv}$. Note also that from the practical point of view (with realistic number of control points and size of MLC discretization matrix) $\sum_{k\in K, i\in I, j\in J} \bar{M} D^k_{ijv} >> U_v$.
- $\bar{d} \leq U_v$, $\forall v \in V_t$. Otherwise, $x_v = 0$ and the variable can be eliminated from the problem. To keep problem instances more interesting, we further assume $\bar{d} < U_v$.
- $L_v < \bar{d}$, $\forall v \in V_t$. Otherwise, with the current objective function, $x_v = 1$ will hold in any feasible solution.
- $L_v > 0$ and $D^k_{ijv} > 0$, $\forall k \in K, i \in I, j \in J$.

Note that the last assumption is only necessary for the following discussion, in order to ensure that the studied polytopes are non-trivial (e.g., when $D^k_{ijv} = 0$, one can simply eliminate this bixel (i, j) from the problem; on the other hand, having $L_v = 0$ would allow the trivial solution of "origin").

Next, we will look at subproblems that can be analytically studied but can also be extended to other crucial subproblems and hence provide important insight. We use the notation $P_{i\times j\times k\times v}$ to indicate the convex hull of a problem with i rows, j columns, k control

points, and v voxels. We also note that we will omit any indices that have only a single element for the sake of easier readability.

3.1 A Study of the $P_{1\times n\times|K|\times 1}$ Polytope

Using our notation, $P_{1\times n\times|K|\times 1}$ represents the convex hull of the model with a single row, n columns, $|K|$ control points, and a single voxel. In the following discussion, the indices for rows and voxels will be omitted. For MLCs that allows interdigitation, the interleaf constraints can be removed, resulting in single row subproblems. From an analytical point of view, a model with a single voxel captures the basic settings for the optimization problem, where one is to make decisions on MLC apertures and fluence weights for each control point. Hence, our polyhedral study of the $P_{1\times n\times|K|\times 1}$ polytope will provide us with insights on inequalities that are promising for the general model with multiple voxels. The notation $D^k_{\ell,r} = \sum_{j=\ell}^{r} D^k_j$ for $k \in K$ will be used throughout this section. Also, note that $|\mathcal{L}| = \frac{(n+1)(n+2)}{2}$. In what follows, for simplicity, we use L to represent L_v, as there is only a single voxel.

Proposition 3.1 *$dim(P_{1\times n\times|K|\times 1}) = |K||\mathcal{L}| + n|K| + 1$, when $\bar{M}D^k_{1,n} \geq L$ for each control point $k \in K$, and $|K| \geq 4$.*

Proof. First, note that there are $|K||\mathcal{L}| + n|K| + |K| + 1$ variables (the $y/\bar{z}/z/x$ variables, respectively) and $|K|$ equations, hence $dim(P_{1\times n\times|K|\times 1}) \leq |K||\mathcal{L}| + n|K| + 1$. In order to show $dim(P_{1\times n\times|K|\times 1}) \geq |K||\mathcal{L}| + n|K| + 1$, we list the following $|K||\mathcal{L}| + n|K| + 2$ affinely independent points, where $\epsilon > 0$ is a sufficiently small number and $\exists \bar{k}, \hat{k} \in K$ with $\bar{k} \neq \hat{k}$. We choose an arbitrary $\tau \in \{0, \ldots, n\}$. We have the following three cases.

Case 1 We have altogether $|K|+2$ points with: $y^k_{(\tau,\tau+1)} = 1$, for all $k \in K\backslash\{\hat{k}\}$; and $y^{\hat{k}}_{(0,n+1)} = 1$;

Case 1a 1 solution with: $z^{\hat{k}} = \bar{z}^{\hat{k}}_{j'} = \frac{L}{D^{\hat{k}}_{(1,n)}}$, for all $j' = 1, \ldots, n$; and 0 otherwise;

Case 1b 1 solution with: $z^{\hat{k}} = \bar{z}^{\hat{k}}_{j'} = \frac{U}{D^{\hat{k}}_{(1,n)}}$, for all $j' = 1, \ldots, n$; $x = 1$, and 0 otherwise;

Case 1c 1 solution with: $z^{\hat{k}} = \bar{z}^{\hat{k}}_{j'} = \frac{L+\epsilon}{D^{\hat{k}}_{(1,n)}}$, for all $j' = 1, \ldots, n$ and 0 otherwise;

Case 1d $|K| - 1$ solutions, one for each $k \in K \setminus \{\hat{k}\}$, with: $z^{\hat{k}} = \bar{z}^{\hat{k}}_{j'} = \frac{L}{D^{\hat{k}}_{(1,n)}}$, for all $j' = 1, \ldots, n$; $z^k = \kappa$, for some $0 < \kappa \leq \bar{M}$, and 0 otherwise.

Case 2 There are $|K|(|\mathcal{L}|-1)$ points, one for each $k \in K$ and for each $(\ell, r) \in \mathcal{L}_k$

$$\mathcal{L}_k = \begin{cases} \mathcal{L} \setminus \{(0, n+1)\} & \text{if } k = \hat{k}; \text{ and} \\ \mathcal{L} \setminus \{(\tau, \tau+1)\} & \text{otherwise,} \end{cases}$$

$y^k_{(\ell,r)} = 1$ holds for $(\ell, r) \in \mathcal{L}_k$; $y^{k'}_{(\tau,\tau+1)} = 1$, for all $k' \in K \setminus \{k, \beta\}$, for $\beta = \bar{k}$ if $k = \hat{k}$, otherwise $\beta = \hat{k}$; $y^\beta_{(0,n+1)} = 1$, $\bar{z}^\beta_{j'} = \frac{L}{D^\beta_{(1,n)}}$, for all $j' = 1, \ldots, n$; $z^\beta = \frac{L}{D^\beta_{(1,n)}}$, where $\beta = \bar{k}$ if $k = \hat{k}$, otherwise $\beta = \hat{k}$; and all other variables equal zero.

Case 3 We have $n|K|$ points, one for each $k \in K$ and each $j = 1, \ldots, n$, given as:
$y^k_{(j-1,j+1)} = 1$, $z^k = \bar{z}^k_j = \frac{L}{nD^k_j}$; $y^\beta_{(0,n+1)} = 1$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise;
$y^{k'}_{(\tau,\tau+1)} = 1$, for all $k' \in K \setminus \{k, \beta\}$ for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise;
$\bar{z}^\beta_{j'} = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for all $j' = 1, \ldots, n$; $z^\beta = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise, and all other variables equal zero.

For $k = \bar{k}$, as we have $\bar{z}^{\bar{k}}_j > 0$, the only way to obtain this is through some linear combination of other vectors with $\bar{z}^{\bar{k}}_j > 0$. However, the only vectors with $\bar{z}^{\bar{k}}_j > 0$ have $y^{\hat{k}}_{(0,n+1)} = 0$, which will not give us $y^{\hat{k}}_{(0,n+1)} = 1$ as needed. Hence these vectors are affinely independent to all of the previously introduced vectors. A similar justification can also be made for the case of $k = \hat{k}$. □

Proposition 3.2 *The following are the trivial facets of $P_{1\times n\times|K|\times 1}$, under the condition that $|K| \geq 4$ and other necessary conditions as indicated next to the constraints:*

(a) $x \geq 0$;

(b) $x \leq 1$ *when* $\bar{M}D^k_{(1,n)} > \bar{d}$, $\forall k \in K$ *and* $\bar{d} < U$;

(c) $y^k_{(\ell,r)} \geq 0$, *for all* $k \in K$, *and all* $(\ell, r) \in \mathcal{L}$;

(d) $\bar{z}^k_j \geq 0$, *for all* $k \in K$ *and all* $j \in \{1, \ldots, n\}$;

(e) $\bar{z}^k_j \leq z^k$, *for all* $k \in K$ *and all* $j \in \{1, \ldots, n\}$.

The proof for Proposition 3.2 is provided in the appendices for the sake of readability. Next, we present key results regarding the inequalities presented in Section 2.3.

Proposition 3.3 *The following are non-trivial facets of $P_{1\times n\times|K|\times 1}$:*

(a) $d_v - L_v \geq (\bar{d} - L_v)x_v$; *and*

(b) $z^k - \bar{z}_j^k \leq \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell\geq j \bigvee r\leq j}} \min\{\bar{M}, \frac{U_v}{D^k_{\ell+1,r-1}}\} y^k_{(\ell,r)}$.

Proof.

(a) As we need to satisfy that $d_v = L_v$ when $x_v = 0$, and $d_v = \bar{d}$ when $x = 1$, we can reuse the vectors presented in the proof of Proposition 3.1 with the following modifications: We remove the vector presented in Case 1(c), and in Case 1(b), we replace U by $\bar{d}$.

(b) We use again the vectors presented in the proof of Proposition 3.1 as a starting point. First, note that we need to construct vectors that satisfy:

$$z^k - \bar{z}_j^k = \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell\geq j \bigvee r\leq j}} \min\{\bar{M}, \frac{U}{D^k_{\ell+1,r-1}}\} y^k_{i(\ell,r)}. \tag{24}$$

Most of the vectors presented in Proposition 3.1 satisfies (24), unless noted in the list of modifications presented below. Equation (24) is satisfied under three cases, which we call as Case (α), Case (β), and Case (γ).

Case (α): We have $y^k_{(\ell,r)} = 1$, for $\ell < j < r$, in which case the right hand side of (24) equals zero. This means that either we have $z^k = \bar{z}_j^k = 0$ (which appear in Case 2 of the proof of Proposition 3.1); or $z^k = \bar{z}_j^k = \lambda$, for $\lambda \neq 0$, (which only happens in Case 3 of Proposition 3.1 with $y^k_{j-1,j+1} = 1$, and we have $z^k = \bar{z}_j^k = \frac{L}{nD^k_j}$).

Case (β): We have $y^k_{(\ell,r)} = 1$, for $j \leq \ell$ or $j \geq r$, and $r = \ell + 1$. In this case, $\bar{z}_j^k = 0$, and the right hand side also equals zero. In all but one case presented in the proof of Proposition 3.1 that concern such $y^k_{(\ell,r)}$, we have $z^k = \bar{z}_j^k = 0$. The only vector with such $y^k_{(\ell,r)}$ and with $z^k \neq \bar{z}_j^k$ appears in Case (1d) of the proof, where $(\ell, r) = (\tau, \tau+1)$, $z^k \neq 0$, but $\bar{z}_j^k = 0$. We simply remove this vector.

Case (γ): We have $y^k_{(\ell,r)} = 1$, for $j \leq \ell$ or $j \geq r$, and $r > \ell + 1$. Again, $\bar{z}_j^k = 0$. In the proof of Proposition 3.1, these cases appeared twice, once in Case 2, and once in Case 3 in the proof of Proposition 3.1. The vectors all have $\bar{z}_j^k = 0$. We simply change the value of z^k from their original value assigned in the respective cases in the proof of Proposition 3.1 to $z^k = \min\{\bar{M}, \frac{U}{D^k_{\ell+1,r-1}}\}$. □

3.2 The Special Case of $P_{1\times n\times 1\times 1}$

We now present our results on a number of non-trivial facet-defining constraints for the special case of single row, a single control point and a single voxel. This very special case is

unlikely to be an insightful subproblem for VMAT, but it is more insightful for IMRT, and is therefore included in the paper for the sake of completeness. In the following discussion, the indices for rows, control points and voxels will be omitted. For ensuring feasibility, we simply exclude closed leaf positions from the following discussion, i.e., $\mathcal{L} := \mathcal{L}\backslash\{(j, j+1)|j = 0, \ldots, n\}$. Let $D_{\ell,r} = \sum_{j=\ell}^{r} D_j$. Note that the results presented in the previous section are still valid and hence not repeated here.

Proposition 3.4 *The inequality*

$$z - \bar{z}_j \geq \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell\geq j \bigvee r\leq j}} \frac{L}{D_{\ell+1,r-1}} y_{\ell,r} \tag{25}$$

is valid for $P_{1\times n\times 1\times 1}$ and dominates (56). Moreover, it is facet-defining for $P_{1\times n\times 1\times 1}$ under the general condition of $\bar{M}D_{j''} > L$, $\forall j'' \in J$.

The following proposition concerns constraints that involve two columns, any arbitrary distinct columns).

Proposition 3.5 *The 2-column inequalities for $P_{1\times n\times 1\times 1}$:*

1. *The 2-column inequality with $j < \widehat{j}$,*

$$\begin{aligned} \bar{z}_j + \bar{z}_{\widehat{j}} - z \leq & \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell<j \bigwedge \widehat{j}<r}} \min\{\bar{M}, \frac{U}{D_{\ell+1,r-1}}\} y_{\ell,r} \\ & - \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell\geq\widehat{j} \bigvee r\leq j \bigvee (\ell\geq j \bigwedge r\leq\widehat{j})}} \frac{L}{D_{\ell+1,r-1}} y_{\ell,r} \end{aligned} \tag{26}$$

   is valid for $P_{1\times n\times 1\times 1}$, and is facet-defining under the general condition $\bar{M}D_{j''} > L$, $\forall j'' \in J$.

2. *The 2-column inequality with $j < \widehat{j}$,*

$$\begin{aligned} \bar{z}_j + \bar{z}_{\widehat{j}} - z \geq & \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell<j \bigwedge \widehat{j}<r}} \frac{L}{D_{\ell+1,r-1}} y_{\ell,r} \\ & - \sum_{\substack{(\ell,r)\in\mathcal{L} \\ \ell\geq\widehat{j} \bigvee r\leq j \bigvee (\ell\geq j \bigwedge r\leq\widehat{j})}} \min\{\bar{M}, \frac{U}{D_{\ell+1,r-1}}\} y_{\ell,r} \end{aligned} \tag{27}$$

   is valid for $P_{1\times n\times 1\times 1}$, and is facet-defining under the conditions that $\bar{M}D_{j''} > L$, $\forall j'' \in J$ and $\exists(\ell, r) \in \mathcal{L}\backslash\{(0, n+1)\}$ such that $\ell \geq j$ or $\widehat{j} \geq r$ and $\bar{M}D_{\ell+1,r-1} \geq U$.

For the proofs, we refer the interested reader to the appendices.

4. Solution Methodologies

In this section, we introduce four heuristic solution methods. We first discuss three natural ways to obtain Lagrangian relaxations to the original MIP. We then explain why only two of them are implemented in Sections 4.2 and 4.3. The third heuristic is based on a different MIP-formulation of the model, and the last heuristic is based on the metaheuristic idea of Variable Neighborhood Search, where we derive some problem-specific features to guide the search. We note that customized exact methods for our MIP model are studied in a companion paper [4].

4.1 Lagrangian Relaxation (LR) for Upper Bounds

Here, we consider three variants:

1. **Relax constraints (3),(4), (50)-(51), (55)-(58)**: As proven by Proposition 4.1, the subproblem is guaranteed to be solved in polynomial time.

2. **Relax constraints (5)-(52)**: This creates K separate subproblems, one for each control point k (each subproblem being an IP problem).

3. **Relax constraints (55) and (57)**: This relaxation generates two subproblems, one for y variables and one for z and x variables.

Let LR_i indicate the Lagrangian Relaxation and LD_i be the optimal solution of the Lagrangian dual for $i = 1, 2, 3$, i.e.,

$$LD_i = \min_{u} \max_{x,y,z,\bar{z}} LR_i(u, x, y, z, \bar{z}),$$

where u is the vector of Lagrangian multipliers.

Proposition 4.1 $LD_1 = \max\{(1)|(2) - (52), (55) - (58), 0 \leq x \leq 1^{|V_t|}, 0 \leq y \leq 1^{|I|\times|\mathcal{L}|}\}$.

In words, optimizing the Lagrangian dual for the first relaxation will generate a bound equal to LP relaxation of the original problem.

Proof. For this subproblem, consider the following network: For each $i \in I$, draw a network with $K+2$ levels as follows: Level 0 has a single, dummy source node with supply of 1, level $K+1$ has a single dummy sink node with demand 1, and each of the other K levels have $\mathcal{L}$ nodes (representing all $(\ell, r) \in \mathcal{L}$) with a demand of zero. First, from the dummy source node,

draw arcs to all $\mathcal{L}$ nodes in level 1. From level 1, draw arcs to level 2 in the way that MLC shape (ℓ, r) in level 1 can be changed to (ℓ', r') in level 2. Repeat this for all levels until level K is reached. Then, connect all the nodes in that level to the dummy sink node. It is easy to observe that solving a shortest path problem through this network, where arc costs will be defined as a combination of original objective function coefficients and Lagrangian multipliers, will generate integral solutions to y variables that satisfy the constraints (2),(5),(6). Finally, note that constraints (52) are independent from previous constraints and there is only a single binary variable on each constraint related to linear variables; therefore all extreme point solutions will be integral. Therefore, $conv(\{(x, y, z)|(2), (5), (6), (48), (49), (52), x \in \{0, 1\}^{V_t}, y \in \{0, 1\}^{|I|\times|\mathcal{L}|}\}) = conv(\{(x, y, z)|(2), (5), (6), (48), (49), (52), 0 \leq x \leq 1^{V_t}, 0 \leq y \leq 1^{|I|\times|\mathcal{L}|}\})$. $\square$

Since the Lagrangian dual for this relaxation does not provide a bound better than the LP relaxation bound of the original problem, we do not investigate this further. However, we make this technical comment for the sake of completeness.

Corollary 4.1 $LD_3 \leq LD_1$.

On the other hand, for LD_2 and LD_3, the subproblems do not in general produce naturally integer solutions, hence the Lagrangian dual bounds they generate will be at least as strong as (or probably lower than) the LP relaxation bounds. Although, this may involve more computational effort. As we will see in numerical results, LR_2 is computationally cheap, whereas LR_3 requires significant effort.

4.2 LR-Based Heuristic I

Our first heuristic is based on the control point independence feature of LR_2. We relax machinery constraints that link the adjacent control points, i.e., (5)- (52), and are thus able to solve each control point subproblem independently. Instead of penalizing all the violated constraints in the objective function like the usual LR methods do, we penalize only the dose violations (i.e., (50)-(51)), but "fix" the neighboring control point machinery constraints ((5), (6)) as follows. Once a single-control point subproblem is solved and its MLC shape is determined, i.e., y-variables are *fixed*, we can impose the machinery constraints for the neighboring control points. This is done iteratively until all MLC shapes for all control points are obtained. Then, the problem reduces to a simple IP without machinery constraints (it would have been simply an LP if constraint (52) was not included). Algorithm 1 provides an overview of the heuristic, where $(a)^+ = \max(a, 0)$.

Algorithm 1 A heuristic that exploits control points consecutively, generating $|K|$ candidate solutions.

for $k \in K$ **do**
  Solve the problem P1: $\min\{\sum_{v \in V_t}(L_v - d_v)^+ + \sum_{v \in V}(d_v - U_v)^+ | (y, z, \bar{z}) \in X_{rel}\}$
  where $X_{rel} = \{(2) - (4), (53), (55) - (58)\}$, for only k;
  Solutions are fixed y-variables $\implies$ MLC-shape for control point k;
  for $k' = k + 1 \ldots |K|$ and $k' = k - 1 \ldots 1$ **do**
    Solve the problem P1': $\min\{\sum_{v \in V_t}(L_v - d_v)^+ + \sum_{v \in V}(d_v - U_v)^+ | (y, z, \bar{z}) \in X'_{rel}\}$
    where $X'_{rel} = \{(2) - (49), (53), (55) - (58)\}$, for only k';
    Fix shapes for k';
  end for
  Now all shapes fixed, solve the original problem;
end for

When we minimize the objective function in the inner loop, we involve *all* control points that are fixed so far, in addition to k', in order to minimize the error function $\sum_{v \in V_t}(L_v - d_v)^+ + \sum_{v \in V}(d_v - U_v)^+$ more accurately and to obtain more involved fixing decisions. Also note that one can replace this objective function with a discrete function that counts infeasibilities, i.e., $\min \sum_{v \in V_t} w_v^- + \sum_{v \in V} w_v^+$ such that $w_v^-, w_v^+ \in \{0, 1\}$ and $L_v w_v^- \geq L_v - d_v$; $(\bar{M} \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} D_{ijv}^K) w_v^+ \geq d_v - U_v$. This will simply provide an equal weight for each voxel to satisfy their dosage lower and upper bound constraints; rather than the "error". We believe that in theory, such a counting objective is appropriate, at least mathematically, and is based on a similar logic like the objective function of our problem. However, in a number of preliminary tests we ran, we have observed that this discrete counting function does not behave well computationally possibly due to symmetry, and therefore we use the original error function in the final algorithm.

In the current implementation, once the problem P1 is solved for k, first the forward (i.e., for $k' = k + 1$ to $k' = |K|$) and then the backward (i.e., for $k' = k - 1$ to $k' = 1$) subproblems are solved. Forward and backward subproblems can be solved separately in parallel at the same time if needed for a faster result, although we consider in this paper the sequential time for fair comparison purposes and leave parallelization to future research. We also note that one can apply different running orders of subproblems in the inner loop to obtain different solutions, e.g., in the alternating order (i.e., $k' = k + 1, k' = k - 1, \ldots, k' = K, k' = 1$) or in a randomized order (i.e., pick with probability p forward or probability $1 - p$ backward subproblem). We ran some limited experiments on this aspect and did not observe any significant difference. Hence we implemented the simple sequential order in this paper, although we plan more extensive computational tests as part of future research. The

computational performance of Algorithm 1 (in particular how often feasible solutions can be found, solution qualities, and computation times) will be discussed in Section 5.

4.3 LR-Based Heuristic II

Our second LR-based heuristic is a combination of subgradient optimization and an IP-heuristic, applied in an iterative fashion to the third Lagrangian relaxation (LR_3). If we relax constraints (55) and (57) and assign multipliers α_{ij}^k and β_{ij}^k respectively to these two sets of constraints, we obtain two separate subproblems as follows:

$$\begin{aligned}
(Pr.1): &\max_{x,z,\bar{z}} \sum_{v\in V_t} x_v + \sum_{k\in K}\sum_{i\in I}\sum_{j\in J}(\beta_{ij}^k - \alpha_{ij}^k)\bar{z}_{ij}^k - \beta_{ij}^k z^k \\
&\text{s.t.} \quad (48)-(53),(56),(58); \text{ and} \\
(Pr.2): &\max_{y} \sum_{k\in K}\sum_{i\in I}\sum_{j\in J} \bar{M}(\alpha_{ij}^k - \beta_{ij}^k) \sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} y_{i(\ell,r)}^k \\
&\text{s.t.} \quad (2)-(6).
\end{aligned}$$

Once these independent subproblems are solved individually, we perform subgradient optimization by updating the multipliers using a stepsize θ described as follows:

$$\alpha_{ij}^k \leftarrow \alpha_{ij}^k + \theta\left(\bar{z}_{ij}^k - \bar{M}\sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} y_{i(\ell,r)}^k\right), \text{ and}$$

$$\beta_{ij}^k \leftarrow \beta_{ij}^k + \theta\left((\bar{M}(-1+\sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} y_{i(\ell,r)}^k) + z^k) - \bar{z}_{ij}^k\right).$$

Every time the multipliers are updated, the solution to (Pr.2) is used to fix all y-variables and then a simplified version of the original problem is optimized to obtain a heuristic solution. This algorithm then iterates back to solving the two subproblems with the new multipliers. The computational performance (in particular how often feasible solutions can be found, solution qualities, and computation times) will be discussed in Section 5.

4.4 A Centering-Based Heuristic

The challenge with the original formulation is the vast number of the y variables, as these are defined for any possible left-right leaves combinations. If one knew in advance where the

"center" of an opening in a row lies, then one could simply define n binary variables for this row instead of n^2 binary variables (hence a significant reduction of problem dimension and computational complexity), where these binary variables either indicate the left-leaf position (if column position is smaller than center) or the right-leaf position (if column position is bigger than center). In a heuristic fashion, one can extend this idea to the "center of an opening" for a given control point, i.e., a column being chosen as the center column for *all* rows (rather than defining a center column for each row). Such a intuitive approach has the potential that one can solve this (probably easy) problem iteratively multiple times, e.g., by using different centering schemes. The pseudocode for the proposed heuristic is presented in Algorithm 2.

First, we define our notation and reformulate the problem for control point k. Let y^{ℓ}_{ij} (y^{r}_{ij}) be binary variables for row i and column j, where $y^{\ell}_{ij} = 1$ ($y^{r}_{ij} = 1$) holds when the left (right) leaf position is on column j. Let $\hat{c}^k$ represent the predefined center column for this control point k, i.e., it is taken as center column for all rows, and $\hat{\ell}_i$ ($\hat{r}_i$) indicate the left (right) leaf position fixed for row i of the neighboring control point (if k is not the first

processed control point). The reformulation is then as follows:

$$\sum_{j=0}^{\hat{c}^k} y_{ij}^{\ell} = 1 \qquad \forall i \in I \tag{28}$$

$$\sum_{j=\hat{c}^k+1}^{n+1} y_{ij}^{r} = 1 \qquad \forall i \in I \tag{29}$$

$$d_v = \sum_{i \in I} \sum_{j \in J} \left(\bar{z}_{ij}^k \times D_{ijv}^k \right) \qquad \forall v \in V \tag{30}$$

$$\bar{z}_{ij}^k \leq \bar{M} \sum_{j'=0}^{j-1} y_{ij'}^{\ell} \qquad \forall i \in I,\ \forall j \in [1, \hat{c}^k] \tag{31}$$

$$\bar{z}_{ij}^k \leq \bar{M} \sum_{j'=j+1}^{n+1} y_{ij'}^{r} \qquad \forall i \in I,\ \forall j \in [\hat{c}^k + 1, n] \tag{32}$$

$$\bar{z}_{ij}^k \leq z^k \qquad \forall k \in K,\ \forall i \in I,\ \forall j \in J \tag{33}$$

$$\bar{z}_{ij}^k \geq \bar{M} \left(-1 + \sum_{j'=0}^{j-1} y_{ij'}^{\ell} \right) + z^k \qquad \forall i \in I,\ \forall j \in [1, \hat{c}^k] \tag{34}$$

$$\bar{z}_{ij}^k \geq \bar{M} \left(-1 + \sum_{j'=j+1}^{n+1} y_{ij'}^{r} \right) + z^k \qquad \forall i \in I,\ \forall j \in [\hat{c}^k + 1, n] \tag{35}$$

$$\bar{z}_{ij}^k \geq 0 \qquad \forall i \in I,\ \forall j \in J \tag{36}$$

$$y_{ij}^{\ell} = 0 \qquad \forall i \in I,\ \forall j \in J' \backslash \{\hat{\ell}_i - \delta, \hat{\ell}_i + \delta\} \tag{37}$$

$$y_{ij}^{r} = 0 \qquad \forall i \in I,\ \forall j \in J' \backslash \{\hat{r}_i - \delta, \hat{r}_i + \delta\} \tag{38}$$

One important aspect that needs further elaboration is the selection of $\hat{c}^k$ for each control point k. We propose the following intuitive possibilities:

$$\hat{c}_1^k = \frac{\sum_{v \in V_t} L_v \left(\frac{\sum_{i \in I} \sum_{j \in J} (j - 0.5) D_{ijv}^k}{\sum_{i \in I} \sum_{j \in J} D_{ijv}^k} \right)}{\sum_{v \in V_t} L_v}$$

Algorithm 2 A centering-based heuristic, generating $|K|$ candidate solutions.

Pre-solve to obtain $\hat{c}^k$ for each control point $k \in K$;
for $k \in K$ **do**
  Using $\hat{c}^k$, solve the problem: $\min\{\sum_{v \in V_t}(L_v - d_v)^+ + \sum_{v \in V}(d_v - U_v)^+ | (y^\ell, y^r, z, \bar{z}) \in X_{ref}\}$
  where $X_{ref} = \{(28) - (36)\}$, for only Control point k;
  Fix shapes for k, set $\hat{\ell}_i$ and $\hat{r}_i$ values;
  for $k' = k+1 \ldots |K|$ and $k' = k-1 \ldots 1$ **do**
    Using $\hat{c}^{k'}$, solve the problem: $\min\{\sum_{v \in V_t}(L_v - d_v)^+ + \sum_{v \in V}(d_v - U_v)^+ | (y^\ell, y^r, z, \bar{z}) \in X'_{ref}\}$
    where $X'_{ref} = \{(28) - (38)\}$, for only k';
    Fix shapes for k', set $\hat{\ell}_i$ and $\hat{r}_i$ values;
  end for
  Now all shapes fixed, solve the original problem;
end for

$$\hat{c}_2^k = \frac{\displaystyle\sum_{v \in V_o} \frac{1}{U_v} \left(\frac{\displaystyle\sum_{i \in I} \sum_{j \in J} (j - 0.5)(1/D_{ijv}^k)}{\displaystyle\sum_{i \in I} \sum_{j \in J} (1/D_{ijv}^k)} \right)}{\displaystyle\sum_{v \in V_o} \frac{1}{U_v}}$$

$$\hat{c}_3^k = w_1 \hat{c}_1^k + w_2 \hat{c}_2^k$$

The value $\hat{c}_1^k$ emphasizes higher dose and lower L_v, and depends on only tumor voxel parameters whereas the value $\hat{c}_2^k$ emphasizes lower dose and higher U_v, and depends only on sensitive tissue parameters. $\hat{c}_3^k$ simply combines these two with weights w_1, w_2. We observed minimal differences on some preliminary computational tests using different weights, hence we use $\hat{c}_3^k$ with $w_1 = 0.5 = w_2$ in the final algorithm.

The main advantage of this framework is not only a significant reduction in the number of variables but also in the elimination of inter-leaf constraints. On the other hand, this method has the disadvantage that it limits the candidate opening patterns for a control point to only "centered" patterns (e.g., the pattern cannot be diagonally-shaped) and infeasibilities are more probable. We will discuss these aspects in detail in Section 5. We also note that this heuristic is related to the first LR-based heuristic, where single-control point-problems are solved consecutively for each control point, after the MLC apertures for the previous control points are fixed. The main difference here is that the subproblem is even further simplified, since allowing only one center for a whole control point reduces the solution space

significantly. We will compare computational results for both of these methods in the next section.

Finally, we also note that one might use this approach in an exact fashion, where the parameters $\hat{c}^k$ will need to be redefined as variables. This will require a sophisticated approach using a specialized branch-and-bound and column generation scheme, which is investigated thoroughly in a companion paper [4].

4.5 A Guided Variable Neighborhood Scheme (GVNS)

Our Guided Variable Neighborhood Scheme (GVNS) aims to tackle problems of large scale, and therefore, feasibility is the primary objective; the first attempt is to obtain solutions that satisfy all machinery and dose constraints. When such a feasible solution is found, a pre-determined number of attempts to improve the original objective function will be triggered. This is done by keeping track of the current best feasible solution with the highest original objective value. If there are no improvements to the original objective value after a predetermined number of dose and machinery-feasible solutions are found, we return the current feasible solution with the best original objective value as an output.

The method randomly generates fluence weights and MLC shapes that satisfy the machinery constraints for each control point. We define two main neighborhoods:

- N^μ is the neighborhood of solutions obtained by modifying the fluence weights. This is further divided into N^μ_+ (increase MU by Δ_μ) and N^μ_- (decrease MU by Δ_μ).
- N^s is the neighborhood of solutions obtained by modifying the MLC shapes.

For the former, we further define $N^\mu_{\pm 1} \subset N^\mu_{\pm 2} \subset \cdots \subset N^\mu_{\pm VNS_{max}}$, i.e., $N^\mu_{\pm i}$ is the neighborhood defined by having i MUs increased (or decreased) while satisfying the machinery constraints. (In our experiments, we used $\Delta_\mu = 1$). VNS_{max} denotes the maximum number of MUs that are allowed to be changed, calculated as a fraction of $|K|$.

Similarly, we also have $N^s_1 \subset N^s_2 \subset \cdots \subset N^s_{VNS_{max}}$, with N^s_p representing the random selection of p control points and modifying the associated MLC shapes by moving each leaf by a random selection from the feasible moves of the leaves. Our preliminary experiments indicated that the result of implementing two different shape change schemes (enlarge_shape and reduce_shape) does not differ significantly from that of the random shape change. With the former, while fixing violations in underdosed voxels, overdose is often induced in other voxels, and vice versa. We implemented four variations of our heuristic method:

- **Method 1** - *Random Descent Local Search*

  In this method, the neighbour search alternates between MU change (N^{μ}) and shape change (N^{s}). The method randomly selects a control point and modifies its MU or shape.

- **Method 2** - A *Guided Search*

  This is done in a way that the neighborhood search is "guided" by the current solution to improve machinery constraint satisfaction. Let $oCNT$ be the number of voxels overdosed and $uCNT$ be the number of voxels under-dosed. Then:

  - **if** $\frac{oCNT-uCNT}{oCNT} \geq 0.1$, we search the neighborhood of $N^{\mu}_{-1}$to decrease the MU;
  - **if** $\frac{uCNT-oCNT}{uCNT} \geq 0.1$, we search the neighborhood of N^{μ}_{+1} to increase the MU;
  - **otherwise** we search N^{S} to change the shape.

- **Method 3** - A *Variable Neighborhood Search*

  In this method, we also alternate between MU change and shape change. First, we calculate $VNS_{\max} = \lceil \gamma |K| \rceil$, for $0 < \gamma < 1$ a user-determined value (we used $\gamma = 0.3$ in our experiments). Initially, we set $size_VNS = \lceil \frac{|oCNT-uCNT|}{|V|} \times (VNS_{\max} - 1) \rceil$. If there is no improvement to the objective value, we increase $size_VNS$ by 1, until $VNS_{\max}$ is reached.

  Method 3 is executed in two different ways. In Method 3a, shape change is carried out when the iteration count is even with $size_VNS$ control points changed at a time, and MU change is carried out otherwise. In Method 3b, in each iteration of the neighborhood search, MU change and shape change are executed iteratively.

- **Method 4** - A *Guided Variable Neighborhood Search*

  Our preliminary results showed that Method 4 outperformed Methods 1–3, hence we will describe only this method in details. The VNS and its variations have been around for many years (for general literature overview, see [14, 21]). In our problem-specific implementation, we modified the VNS by integrating the "guided search" idea of Method 2.

  We begin with a randomly generated machinery feasible solution with leaf positions and fluence weights that satisfy the machinery constraints. We then calculate the dose

delivered to each voxel to find out the dose violations, and perform a search guided in the manner of promoting dose satisfaction.

The *objective function* is given by $vCNT = oCNT + uCNT$. If $vCNT > 0$, the GVNS will proceed to search for "neighboring" solutions by using one of the two neighborhood search schemes N^{μ} and N^{s}. Our method attempts to increase or decrease MUs in the first instance. We also perform MLC shape change on a regular basis, or if no consecutive-control point-feasible MU change is possible. Should there be no machinery feasible leaf movements available, we modify our shape change procedure by first selecting i control points randomly, select the left- and right-leaf positions for the *first row* of the MLC of each of these i control points, and generate leaf positions for subsequent rows that satisfy the inter-leaf constraints. It is possible that some consecutive control point leaf restrictions are violated, and these violations are added into the cost function. Once in a while, a new MLC-feasible solution will be generated for all $|K|$ control points, allowing random starts for the search. See Algorithm 3 for the pseudocode. We note the key notation, as follows:

- $ItCNT$ a counter for number of iterations the GVNS has been performed.
- $ItMAX$ a predetermined maximum number of iterations to be performed.
- $ItNEW$ a new machinery feasible solution is generated every $ItNEW$ iterations.
- κ a counter for the number of occurrences when a feasible solution is found and it does not improve the current best original objective function value.
- $\kappa_{\max}$ the maximum value for κ allowed in the GVNS (in our experiments, $\kappa_{\max} = 5$).
- $\omega(x')$ the value of the original objective function given by solution x'.
- ω^{*} best value of original objective function out of all feasible solutions found.

Algorithm 3 GVNS for VMAT, finding feasible solutions (or a solution with least violations).

Set $ItCNT = \kappa = \omega^* = 0$; Randomly generate a machinery feasible *initial solution* x';
Set objective cost $z' = oCNT + uCNT$; Set *best soln* $x^* \leftarrow x'$, *best cost* $z^* \leftarrow z'$;
if $z' = 0$ **then** set $\omega^* = \omega(x')$, $\kappa \leftarrow \kappa + 1$;
end if
while $(\kappa < \kappa_{\max}) \ \wedge \ (ItCNT < ItMAX)$ **do**
    if (ItCNT $\mod$ ItNEW $= 0$) **then** Generate a new x';
    end if
    if $(oCNT > uCNT) \vee (uCNT > oCNT)$ **then**
        Set $sizeVNS = \left\lceil \frac{|(oCNT - uCNT)|}{|V|}(VNS_{max} - 1) \right\rceil$;
        if $oCNT > uCNT$ **then**
            Reduce MUs ;
            Obtain x'' by random $N^{\mu}_{-sizeVNS}(x')$, calculate $oCNT$, $uCNT$; set $z'' \leftarrow oCNT + uCNT$;
            if $z'' = 0$ **then**
                if $\omega(x'') \geq \omega^*$ **then**
                Set $\omega^* \leftarrow \omega(x'')$ and Set $\kappa = 0$;
                else
                Set $\kappa \leftarrow \kappa + 1$;
                end if
            else if $z'' > z^*$ **then**
                Set $sizeVNS \leftarrow \min\{VNS_max, sizeVNS + 1\}$;
            else if $z'' < z^*$ **then**
                Set $z^* \leftarrow z''$ and $x^* \leftarrow x''$;
                Set $sizeVNS \leftarrow \max\{1, sizeVNS - 1\}$;
            end if
        else if $uCNT > oCNT$ **then**
            Increase MUs;
            Obtain x'' by random $x'' \in N^{\mu}_{+sizeVNS}$, calculate $oCNT$, $uCNT$; set $z'' \leftarrow oCNT + uCNT$;
            if $z'' = 0$ **then**
                if $\omega(x'') \geq \omega^*$ **then**
                Set $\omega^* \leftarrow \omega(x'')$ and Set $\kappa = 0$;
                else
                Set $\kappa \leftarrow \kappa + 1$;
                end if
            else if $z'' > z^*$ **then**
                Set $sizeVNS \leftarrow \min\{VNS_max, sizeVNS + 1\}$;
            else if $z'' < z^*$ **then**
                Set $z^* \leftarrow z''$ and $x^* \leftarrow x''$;
                Set $sizeVNS \leftarrow \max\{1, sizeVNS - 1\}$;
            end if
        else if No feasible MU change can be made **then**
            Perform Change_Shape routine;
            Obtain x'' by random $x'' \in N^{s}_{sizeVNS}(x)$, calculate $oCNT$, $uCNT$; set $z'' \leftarrow oCNT + uCNT$;
            if $z'' = 0$ **then**
                if $\omega(x'') \geq \omega^*$ **then**
                Set $\omega^* \leftarrow \omega(x'')$ and Set $\kappa = 0$;
                else
                Set $\kappa \leftarrow \kappa + 1$;
                end if
            else if $z'' > z^*$ **then**
                Set $sizeVNS \leftarrow \min\{VNS_max, sizeVNS + 1\}$;
            else if $z'' < z^*$ **then**
                Set $z^* \leftarrow z''$ and $x^* \leftarrow x''$; and $sizeVNS \leftarrow \max\{1, sizeVNS - 1\}$;
            end if
        end if
    else
        Perform Change_Shape routine (as described above);
    end if
    $ItCNT \leftarrow ItCNT + 1$;

5. Numerical Results

In this section, we present our numerical results for the various heuristic methods, starting with a discussion of some preliminary aspects. It would be ideal if we were able to test the methods we developed on real clinical data. Unfortunately, we currently do not have access to an in-house dose deposition coefficient (D^k_{ijv}) calculation software such as the one used in [25]. Moreover, while we do have some DICOM images available, converting these images into indexed voxels with associated structure label and dose requirement labels would involve a substantial amount of work without any in-house tools. In spite of these difficulties, we designed a problem generator that generates problem instances as close to the reality as possible. For our experiments, we first used randomly generated problem instances that are feasible (30 instances with sizes from small to large, plus 3 very large instances) for the testing of all our solution methods. This is described in detail in [3], where documentation and the entire data set are available for the benefit of the research community, potentially providing a common platform for researchers to experiment and benchmark different algorithms and methods. We also generated seven extra large problem instances with the number of non-zero D^k_{ijv} values comparable (in some instances larger) to those clinical instances reported in the recent work of [25]. In this respect, we believe that the extra large problem instances can represent realistic clinical cases. We tested these extra large problem instances only with the GVNS, as the IP-based heuristics were not able to solve problems of this size. Due to the size of data, this extra large data set are available upon request.

Table 1 Parameters used in the problem generator.

Parameters	Description
m/n	number of rows/columns in MLC
$w/h/d$	width/height/depth of treatment field (# voxels)
$\|K\|$	number of control points
ρ	for generating U_v and L_v by using $d_v(1 \pm \rho)$; tested values $\rho = 0.08,\ 0.1,\ 0.2$
$z_{\max}$	max. value for MU allowed from any control point
Δ	max. MU change in consecutive control points
δ	max. leaf traveling in consecutive control points
$d_{\max}$	max. possible value for D^k_{ijv}

In Table 1, we provide an overview of the parameters used in the problem generator. This notation is used to name instances in the format of MLC-Voxels/$|K|$-ρ-z_{max}-Δ-δ-d_{max},

e.g., 6-6-6-6-6/16-0.1-5-2-2-3 is an instance with a 6×6 MLC bixels and $6 \times 6 \times 6$ voxels, 16 control points, and so on.

All computations reported in this section are executed on a PC with i7-2600 processor (3.40 GHz) and 8GB allocated memory. The optimization models, Lagrangian relaxations (Section 4.1), and all the IP-based heuristics (Sections 4.2-4.4) are implemented in OPL modeling language and tested using IBM ILOG Cplex 12.2, whereas the GVNS heuristic (Section 4.5) is implemented in C++. Before discussing results comparing the efficiency of the proposed heuristic methods, we discuss some key preliminary tests.

With the alternative formulation mentioned at the end of Section 1.2 and presented in detail in the appendices, we have run some tests for the sake of comparing the effectiveness of different formulations in different heuristics. The results are presented in Table 2, where LANG refers to the alternative formulation (Langer-type) and AMT refers to our own formulation from Section 2. The problem instances are named with the MLC size and number of control points ($m \times n \times |K|$). The columns labelled "Time" are the total computation time in seconds, and the columns labelled "Obj" are the values of the original objective function value $\max \sum_v x_v$.

Table 2 Using Langer-type formulation (LANG) versus using our formulation (AMT) for HEUR 1 and HEUR 2. Time in seconds.

Problem	HEUR1				HEUR2			
	LANG		AMT		LANG		AMT	
$(m \times n \times \lvert K\rvert)$	Time	Obj	Time	Obj	Time	Obj	Time	Obj
$6 \times 6 \times 16$	67.39	7	77.97	9	2545.92	17	131.91	13
$7 \times 7 \times 8$	22.20	1	20.97	0	2918.89	60	109.03	14
$8 \times 8 \times 8$	520.65	9	2090.94	13	28.38	0	57.74	0
$12 \times 12 \times 10$	349.64	4	1517.79	13	20.97	0	32.29	0

We can see that in some cases, the Langer-type model is computationally more efficient, but the solution produced is of poorer quality. In other cases, our model is computationally more efficient, but the solution produced is not as good as those from the Langer-type model. In general, the longer the computation time, the higher the chance a better solution is found. Since neither model seemed to be clearly superior to the other, we decided to use the model we described in detail for all HEUR1 and HEUR2 experiments.

Our first set of experiments is the computational complexity of the test instances. Out of 33 problems, ILOG Cplex with default settings and 1hour/1% relative gap limitation could find feasible solutions for only 8 problems, proving optimality for only 2 of these. (We

emphasise again that a feasible solution is both dose and machinery feasible, and is therefore a feasible treatment plan). These results are presented in Table 3, where the t/gap column indicates time if < 3600 sec. and otherwise $gap = \frac{(UB-LB)}{LB}$. For instances with 15x15 MLC arrays and above (except 15-15-15-15-15/8-0.1-5-2-3-3), Cplex could not even find an upper bound in this time (see appendices for details).

Table 3 Default Cplex runs with solutions found.

Problem instance	LB	UB	t/gap
6-6-6-6-6/16-0.1-5-2-2-3	17	86	405.88%
7-7-7-7-7/8-0.08-5-1-1-3	273	274	801
7-7-7-7-7/8-0.1-5-1-1-3	103	118	14.56%
7-7-7-7-7/8-0.2-5-1-1-3	6	33	450.00%
7-7-7-7-7/12-0.2-5-1-1-3	0	92	∞
8-8-8-8-8/16-0.1-5-2-2-3	27	426	1477%
12-12-12-12-12/16-0.1-3-2-3-2	1728	1728	993
12-12-12-12-12/16-0.1-5-3-4-3	312	1710	448.08%

Next, we discuss the computational complexity of the Lagrangian relaxations presented in Section 4.1. For 3 groups of test instances with MLC sizes 7×7, 12×12 and 15×15, we ran 100 iterations of each Lagrangian relaxation on each test instance with the Lagrangian multipliers initialized using randomly selected values and updated as described in Section 4.3. The average computational time over all test instances for a single iteration of each relaxation is given in Table 4, where * indicates that there were iterations that were not finished before a pre-set time limit of 900 seconds.

Table 4 Average times (in sec.) for Lagrangian relaxations.

MLC size	LR_1	LR_2	LR_3
7×7	0.81	0.51	23.64
12×12	15.48	6.05	511.21*
15×15	64.56	17.04	628.50*

As these results show, LR_3 is significantly harder to solve (not even finishing for many iterations of bigger problems when $cplex.tilim = 900$ is used), and as we will see later, this affects the efficiency of the second LR-based heuristic. On the other hand, LR_2 can be solved very efficiently, even faster than LR_1. Note that one might improve the computational times of LR_1 by implementing the network idea discussed in the proof of Proposition 4.1.

Finally, we present and discuss the computational results for the four proposed methods. We use the abbreviations *LRHeur1*, *LRHeur2* and *CentHeur* to refer to the **LR**-based

Heuristics (Sections 4.2 and 4.3) and **Cent**ering-based **Heur**istic (Section 4.4), respectively. To ensure reasonable efficiency, pre-set time limits were used for different routines of all the IP-based heuristics, for which we run extensive preliminary tests to determine these parameters. In order to avoid fine tuning and obtain fair comparisons, we used the same time limit of *cplex.tilim* = 90 for all different subroutines except the "fixed heuristic" executed at the end of each iteration (i.e., the heuristic that uses the fixed y variables to solve the original problem). The fixed heuristic uses a time limit of *cplex.tilim* = 120 to increase the probability of finding feasible solutions. We note that increasing these time limits does not necessarily improve solution quality, as results in the appendices show for the case of *cplex.tilim* = 900 (and *cplex.tilim* = 1200 for the fixed heuristic). Finally, we note that for *LRHeur2*, we set the step size $\theta = 0.01$ (based on preliminary testing with a range of values) and limit the number of subgradient optimization iterations to 20 (based on the fact that LR_3 is computationally hard to solve).

Table 5 Comparing four heuristic methods.

Problem Set	# best overall solutions			
(max MLC size)	*LRHeur1*	*LRHeur2*	*CentHeur*	GVNS
Small (8×8)	4	8	1	-
Medium (12×12)	6	-	4	-
Large (15×15)	7	-	6	-
Very Large	-	-	1	2
Problem Set	# instances with no solutions			
(max MLC size)	*LRHeur1*	*LRHeur2*	*CentHeur*	GVNS
Small (8×8)	-	1	2	7
Medium (12×12)	-	8	-	-
Large (15×15)	-	9	-	-
Very Large	3	3	1	-

Detailed computational results can be seen in Tables 6 and 7. Table 6 is organized in the following column order: objective values of the 1st and best solutions, computation time (in seconds) to the 1st solution, best solution, and total time. Note that since GVNS heuristic has an element of randomness in it and that it is much faster than the other methods, 5 runs are executed for each instance, and details for each of these runs are presented for the sake of completeness. Table 7 presents the results for the IP-based heuristics, where for each heuristic, number of iterations (to the 1st solution, to the best solution, and total number), number of feasible solutions obtained, objective values of the 1st and best solutions, and computation time (in seconds if < 3600, otherwise in the **h**our/**m**inute format) are presented,

in this order. As summarized in Table 5, *LRHeur2* is very efficient for small problems but fails to find feasible solution even for most of the medium problems. It is also important to note that *CentHeur* and GVNS are quite inefficient for small problems (they either fail or find poor solutions), which is an expected result as these methods are prone to infeasibility when the search space is limited. *LRHeur1* and *CentHeur* seem to be similarly efficient for medium and large problems (especially the difference between their best solutions is less than 10% for all large instances). Finally, we emphasize that the GVNS is the method of choice when the problem sizes become very large that IP-based heuristics mostly fail or cannot generate good solutions.

As the results indicate, *LRHeur1* seems to be the most successful method overall for the aspect of consistency in finding solutions, having failed to find a feasible solution only in 12 iterations from overall 452 iterations executed (note this excludes the very large instances). A similar comment can be made for GVNS, unless the problem is small. The method finds a solution in every run for all instances with MLC size 8×8 or above, although the best solution qualities might vary significantly from one run to another. However, this variation is encompassed by a very crucial advantage of GVNS: The method runs very fast for the majority of instances. No medium or large instance takes more than one hour for 5 runs (indeed, 14 of these instances take 10 minutes or less), and this is even true for 2 of the 3 very large instances. We also note that the variation in solution time is in general acceptable. IP-based heuristics are far slower, with the fastest one (*CentHeur*) achieving computational times of around 1 hour for the majority of medium and large instances (with occasional variation), and around 2 hours for very large instances. However, we note that in case of limited time, one can run these heuristics for a few iterations only, possibly focusing on control points/iterations that seem to be most successful. For example, *CentHeur* seems more successful for later control points and *LRHeur1* is more successful for earlier control points (although it is also important to note for these two heuristics that like in any other heuristics, there is a factor of chance in this as well as the effect of control parameters used such as time and memory limits).

Table 6 GVNS results detailing all 5 runs for each instance.

Problem Instance	Objective Value 1st sol	Objective Value Best	Time 1st sol	Time Best sol	Time Total	Problem Instance	Objective Value 1st sol	Objective Value Best	Time 1st sol	Time Best sol	Time Total
6-6-6-6-6/16-0.1-5-2-2-3	1	1	80	80	251	12-12-12-12-12/	1	43	2	13	33
	No feasible solution found in 4 runs					24-0.1-5-2-3-3	245	742	5	17	36
7-7-7-7-7/8-0.08-5-1-1-3	13	28	1	95	166		274	274	7	7	27
	No feasible solution found in 4 runs						44	150	1	4	22
							83	563	8	34	56
7-7-7-7-7/8-0.1-5-1-1-3	1	1	42	42	170	15-15-15-15-15/	2	205	2	20	52
	No feasible solution found in 4 runs					8-0.1-5-2-3-3	12	304	4	26	66
7-7-7-7-7/8-0.2-5-1-1-3	No feasible solution found in 5 runs						2	16	1	11	49
7-7-7-7-7/12-0.08-5-1-1-3	No feasible solution found in 5 runs						28	28	1	1	21
7-7-7-7-7/12-0.1-5-1-1-3	No feasible solution found in 5 runs						30	30	6	6	34
7-7-7-7-7/12-0.2-5-1-1-3	No feasible solution found in 5 runs					15-15-15-15-15/	107	1054	3	220	351
7-7-7-7-7/16-0.08-5-1-1-3	No feasible solution found in 5 runs					16-0.1-8-2-3-5	75	212	21	36	196
7-7-7-7-7/16-0.1-5-1-1-3	No feasible solution found in 5 runs						78	130	42	161	306
7-7-7-7-7/16-0.2-5-1-1-3	No feasible solution found in 5 runs						23	118	45	58	202
							692	692	32	32	136
8-8-8-8-8/	6	11	140	266	364	15-15-15-15-15/	76	891	10	111	194
16-0.1-5-2-2-3	1	4	162	234	366	16-0.08-5-2-3-3	358	456	11	135	191
	9	11	20	32	243		168	174	3	27	100
	4	4	145	145	367		33	788	5	37	120
	5	7	126	183	359		1	997	3	57	148
10-10-10-10-10/	68	68	4	4	20	15-15-15-15-15/	1	3101	2	85	144
32-0.3-10-3-3-5	21	21	2	2	19	16-0.1-3-2-3-2	2	2763	5	40	86
	1	1	22	22	34		2277	3024	2	31	95
	45	45	13	13	25		1	2773	9	87	147
	3	3	11	11	24		10	2881	4	23	91
12-12-12-12-12/	1	12	1	169	381	15-15-15-15-15/	31	213	20	71	127
8-0.1-5-2-3-3	4	42	13	42	58	16-0.1-5-1-2-3	558	558	44	44	90
	1	107	10	16	41		261	261	31	31	94
	8	80	7	18	40		13	147	6	25	104
	10	10	4	4	31		14	729	31	113	177
12-12-12-12-12/	376	376	13	13	33	15-15-15-15-15/	1082	1082	23	23	90
16-0.1-3-2-3-2	8	1319	17	42	58	16-0.1-5-2-3-3	489	489	3	3	52
	3	1627	1	8	24		433	1495	18	158	201
	246	246	8	8	22		2392	2392	10	10	63
	414	414	1	1	16		1	667	4	56	103
12-12-12-12-12/	9	326	4	46	70	15-15-15-15-15/	258	2691	17	63	107
16-0.08-5-2-3-3	13	346	1	21	42	16-0.1-5-3-5-3	58	2775	19	37	78
	74	74	1	1	49		1	2069	3	29	78
	15	299	4	47	99		2	2979	10	73	128
	62	128	9	50	87		2428	2428	17	17	60
12-12-12-12-12/	79	79	13	13	33	15-15-15-15-15/	39	39	1	1	47
16-0.1-8-2-3-5	35	35	5	5	19	16-0.2-5-2-3-3	33	33	66	66	103
	21	21	3	3	21		479	479	99	99	140
	22	22	11	11	27		7	7	207	207	249
	10	68	28	58	77		41	41	2	2	45
12-12-12-12-12/	1	26	15	181	374	15-15-15-15-15/	116	373	26	132	210
16-0.1-5-1-2-3	9	75	7	731	1080	24-0.1-5-2-3-3	2194	3195	14	44	111
	57	67	35	173	339		2325	3301	5	93	183
	4	78	7	46	153		141	3217	43	122	205
	7	32	37	56	384		2583	2583	19	19	104
12-12-12-12-12/	1	95	20	51	97	18-18-18-18-18/	3	116	41	217	401
16-0.1-5-2-3-3	13	78	2	45	104	16-0.1-5-3-4-3	71	71	112	112	326
	9	204	4	63	114		13	60	57	154	303
	38	38	16	16	58		717	956	115	148	305
	8	70	12	91	135		582	582	50	50	230
12-12-12-12-12/	6	8	54	176	303	20-20-20-20-20/	5413	5413	129	129	385
16-0.1-5-3-4-3	1	33	61	292	463	16-0.1-5-3-4-3	2454	2454	112	112	370
	4	4	8	8	137		3	7714	149	443	716
	6	32	8	88	283		2	7992	98	582	828
	11	17	29	54	279		7592	7592	186	186	437
12-12-12-12-12/	5	5	42	42	57	25-25-25-25-25/	203	15613	41	8893	66383
16-0.2-5-2-3-3	1	1	8	8	19	16-0.1-5-3-5-3	13936	15606	26	249	2051
	31	31	2	2	17		4451	15625	292	1418	3009
	1	1	2	2	17		51	15625	56	781	2151
	2	2	41	41	55		15182	15182	265	265	1761

Table 7 Detailed results for *LRHeur1*, *LRHeur2* and *CentHeur*, with *cplex.tilim* = 90 set for subroutines, and *cplex.tilim* = 120 set for the "fixed heuristic" run at the end of each iteration.

Problem	*LRHeur1* Results							*LRHeur2* Results							*CentHeur* Results						
	# iterations				Objective values			# iterations				Objective values			# iterations				Objective values		
Instance	1st sol	Best sol	Total	# sol's	1st sol	Best sol	Time	1st sol	Best sol	Total	# sol's	1st sol	Best sol	Time	1st sol	Best sol	Total	# sol's	1st sol	Best sol	Time
6-6-6-6-6/16-0.1-5-2-2-3	1	4	16	16	3	8	75	1	7	20	18	4	15	217	6	15	16	11	2	7	33
7-7-7-7-7/8-0.08-5-1-1-3	1	1	8	8	66	66	47	3	19	20	9	54	61	168	6	8	8	3	13	31	14
7-7-7-7-7/8-0.1-5-1-1-3	1	4	8	8	0	1	19	2	2	20	12	7	7	85	4	4	8	5	0	0	11
7-7-7-7-7/8-0.2-5-1-1-3	1	1	8	8	0	0	812	2	2	20	19	0	0	32	2	2	8	7	0	0	76
7-7-7-7-7/12-0.08-5-1-1-3	1	1	12	5	0	0	362	No feasible solution after 20 iterations							No feasible solution after 12 iterations						
7-7-7-7-7/12-0.1-5-1-1-3	1	6	12	12	3	9	311	2	3	20	19	14	24	76	3	5	12	10	1	3	140
7-7-7-7-7/12-0.2-5-1-1-3	1	1	12	12	0	0	880	1	3	20	20	0	5	81	1	1	12	12	0	0	117
7-7-7-7-7/16-0.08-5-1-1-3	1	8	16	16	4	7	145	3	7	20	10	2	3	165	6	7	16	16	0	1	46
7-7-7-7-7/16-0.1-5-1-1-3	1	1	16	11	2	2	809	3	3	20	1	3	3	77	No feasible solution after 16 iterations						
7-7-7-7-7/16-0.2-5-1-1-3	1	1	16	16	0	0	1218	1	11	20	20	0	1	75	1	1	16	16	0	0	198
8-8-8-8-8/16-0.1-5-2-2-3	1	3	16	16	35	65	3350	3	8	20	18	41	77	1h16m	4	14	16	13	4	56	2782
10-10-10-10-10/32-0.3-10-3-3-5	1	12	32	32	30	688	2h45m	1	1	20	1	0	0	1h21m	3	32	32	30	0	778	2h06m
12-12-12-12-12/8-0.1-5-2-3-3	1	2	8	8	149	230	2159	1	1	20	1	162	162	1h23m	2	8	8	7	65	191	1587
12-12-12-12-12/16-0.1-3-2-3-2	1	1	16	16	1728	1728	1768	No feasible solution after 20 iterations							3	8	16	14	1084	1728	1344
12-12-12-12-12/16-0.08-5-2-3-3	1	2	16	16	1154	1344	1h24m	No feasible solution after 20 iterations							3	14	16	14	244	1330	3539
12-12-12-12-12/16-0.1-8-2-3-5	1	4	16	16	336	491	1h26m	No feasible solution after 20 iterations							2	6	7*	5	182	1360	3561
12-12-12-12-12/16-0.1-5-1-2-3	1	5	16	16	197	325	2h16m	No feasible solution after 20 iterations							2	15	16	15	75	276	3h23m
12-12-12-12-12/16-0.1-5-2-3-3	1	4	16	16	189	341	2h09m	No feasible solution after 20 iterations							2	16	16	13	47	282	4h45m
12-12-12-12-12/16-0.1-5-3-4-3	1	3	16	16	706	768	1h24m	No feasible solution after 20 iterations							2	16	16	15	36	846	1h07m
12-12-12-12-12/16-0.2-5-2-3-3	1	1	16	16	567	567	1h25m	No feasible solution after 20 iterations							2	12	16	15	71	307	1h04m
12-12-12-12-12/24-0.1-5-2-3-3	1	1	24	24	1728	1728	1h57m	No feasible solution after 20 iterations							3	21	24	22	585	1728	1h11m
15-15-15-15-15/8-0.1-5-2-3-3	1	4	8	8	1889	2213	2831	No feasible solution after 20 iterations							2	8	8	7	84	2022	1533
15-15-15-15-15/16-0.1-8-2-3-5	1	5	16	16	2527	2906	3h12m	No feasible solution after 20 iterations							2	16	16	15	155	2948	5h29m
15-15-15-15-15/16-0.08-5-2-3-3	1	2	16	16	3324	3352	2h42m	No feasible solution after 20 iterations							2	16	16	15	1287	3334	1h16m
15-15-15-15-15/16-0.1-3-2-3-2	1	1	16	16	3375	3375	1h42m	No feasible solution after 20 iterations							2	11	16	15	2169	3375	2272
15-15-15-15-15/16-0.1-5-1-2-3	1	3	16	16	3103	3172	2h36m	No feasible solution after 20 iterations							2	16	16	15	1212	3020	1h17m
15-15-15-15-15/16-0.1-5-2-3-3	1	3	16	16	3374	3375	2h	No feasible solution after 20 iterations							2	14	16	15	695	3375	2739
15-15-15-15-15/16-0.1-5-3-5-3	1	1	16	16	3375	3375	1h47m	No feasible solution after 20 iterations							3	11	16	14	203	3375	2754
15-15-15-15-15/16-0.2-5-2-3-3	1	5	16	16	3093	3152	2h39m	No feasible solution after 20 iterations							2	16	16	15	733	3158	1h17m
15-15-15-15-15/24-0.1-5-2-3-3	1	1	16	16	3375	3375	3h18m	No feasible solution after 20 iterations							4	7	24	21	3261	3375	1h06m
18-18-18-18-18/16-0.1-5-3-4-3	No feasible solution after 16 iterations							No feasible solution after 20 iterations							2	9	16	8	2672	5435	1h48m
20-20-20-20-20/16-0.1-5-3-4-3	No feasible solution after 16 iterations							No feasible solution after 20 iterations							3	4	16	2	0	7223	2h16m
25-25-25-25-25/16-0.1-5-3-5-3	No feasible solution after 16 iterations							No feasible solution after 20 iterations							No feasible solution after 16 iterations						

Table 8 Large-scale problem instances. Our GVNS attempt to achieve dose feasibility in the first instance. The time for "1st FEA records the time the first dose and machinery feasible solution is found. "OPT solution refers to the solution that maximizes the number of voxels receiving a desired dose $\bar{d}$.

Problem Instances	Problem characteristics			Solution details			
				1st FEA		OPT	
$(m \times n \times w \times h \times d \times \lvert K \rvert)$	# voxels	# beamlets	# $D^k_{i,j,v} > 0$	Time (sec)	# ITER	Time (sec)	# ITER
$20 \times 20 \times 20 \times 20 \times 20 \times 32$	8,000	12,800	$102,400,000$	34	3	577	56
$10 \times 10 \times 25 \times 25 \times 40 \times 45$	25,000	4,500	$112,500,000$	85	5	1,812	124
$10 \times 10 \times 15 \times 15 \times 30 \times 180$	6,750	18,000	$121,500,000$	470	32	2,859	189
$20 \times 20 \times 20 \times 20 \times 20 \times 45$	8,000	18,000	$144,000,000$	146	8	3,445	184
$10 \times 10 \times 25 \times 25 \times 40 \times 60$	25,000	6,000	$150,000,000$	351	18	78,097	328
$20 \times 20 \times 20 \times 20 \times 20 \times 60$	8,000	24,000	$192,000,000$	289	11	1,804	72
$20 \times 20 \times 20 \times 20 \times 20 \times 72$	8,000	28,800	$230,400,000$	352	11	2,056	67

† Best solution with original objective recorded 3 times before 1,000 iterations reached.

Finally, we also note that only *LRHeur2* and GVNS are methods with a user-defined number of iterations, as it is not dependent on the number of control points (hence more iterations can be added if computational resources are available) to possibly improve solutions.

To summarize, we observe that the GVNS performs very efficiently for larger problems, finding feasible (and many times, good quality) solutions in very short times, where solution times remain fairly stable for different runs of the same problem instance. This justifies the GVNS to be implemented in real-life applications, compared to other methods. In order to verify this conclusion with more realistic problems, we tested the GVNS on 7 problem instances of very large scales which, from computational optimisation point of view, are comparable or even harder than the instances presented in [25], where the most difficult problem instance has 114315187 non-zero $D^k_{i,j,v}$. [25] have presented impressive computational results with GPU computing implemented therein, which achieve very fast computation times in general. However, we note that their algorithm uses an objective function to achieve feasibility that is defined as hard constraints in our model, which makes our model more challenging. We note that their algorithm achieves results, where the dose constraints for most structures are reasonably satisfied, except for the bladder. We present the results for 1st and overall solutions in Table 8, including problem characteristics in the same format as that of [25] for easier comparability. For these runs, we used a quad-core iMac with 2.93GHz Intel Core i7 and 32 GB RAM.

6. Conclusions

In this paper, we studied the combinatorial optimization problem underlying the treatment planning optimization for three treatment modalities: VMAT, TomoTherapy, and CyberKnife. Our contribution lies in presenting a unified mixed integer linear programming formulation for these treatment modalities, investigating some theoretical properties of key subproblems of it, and proposing four solution methods that are also extensively tested using a set of problem instances with sizes ranging from small to very large and practical. We also make our set of test problems available to other researchers, in the hope that this will provide a venue for more efficient comparisons and fruitful discussions in the research community.

One of the important aims we stated at the start was to find feasible solutions that satisfy both the machinery and dose constraints. Our computational results indicated that this was achieved in general, in particular for our most efficient method, Guided Variable Neighbourhood Search, with success in problems of sizes comparable to real clinical problems. One area we left for future research is to investigate possibilities of mixing these different methods such that parallelisation can help practitioners to obtain even better performance. It is also important to note that exact methods and theoretical properties should be investigated further to help us understand these problems better, some of which we address in a companion paper [4]. Further, other variety of objective function and constraint set combinations can be considered for future research, for example, including the volume constraints on OARs and PTVs.

References

[1] Cyberknife m6 series brochure, 2012. Available at `http://www.accuray.com/sites/default/files/ck_m6_series_brochure_en_501004b.pdf`. Last accessed on 03/11/2014.

[2] Tomotherapy H series brochure, 2012. Available at `http://www.accuray.com/sites/default/files/501014.A_TomoTherapy-Brochure.pdf`. Last accessed on 03/11/2014.

[3] K. Akartunalı and V. Mak-Hau. VMAT random problem generator, 2012. Available at `http://www.deakin.edu.au/~vicky/VolArcDat.html`. Last accessed on 03/11/2014.

[4] K. Akartunalı and V. Mak-Hau. Treatment planning optimization for volumetric modulated arc therapy (VMAT): Exact solution approaches. Working Paper, 2014.

[5] H. Cambazard, E. O'Mahony, and B. O'Sullivan. Hybrid methods for the multileaf collimator sequencing problem. In A. Lodi, M. Milano, and P. Toth, editors, *CPAIOR*, volume 6140 of *LNCS*, pages 56–70. Springer, 2010.

[6] D. Craft and T. Bortfeld. On the tradeoff between treatment time and plan quality in rotational arc radiation delivery, 2009. Available at `http://arxiv.org/pdf/0910.4934`. Last accessed on 03/11/2014.

[7] M. Ehrgott, Ç. Güler, H. W. Hamacher, and L. Shao. Mathematical optimization in intensity modulated radiation therapy. *Ann. Oper. Res.*, 175(1):309–365, 2010.

[8] M. Ehrgott and A. Holder. Operations research methods for optimization in radiation oncology, 2009. Available at `http://holderfamily.dot5hosting.com/aholder/research/papers/OROncology.pdf`. Last accessed on 03/11/2014.

[9] Elekta infinity brochure, 2011. Available at `http://www.elekta.com`. Last accessed on 03/11/2014.

[10] A. Fogliata, L. Cozzi, A. Clivio, A. Ibatici, P. Mancosu, P. Navarria, G. Nicolini, A. Santoro, E. Vanetti, and M. Scorsetti. Preclinical assessment of volumetric modulated arc therapy for total marrow irradiation. *Int. J. Radiat. Oncol.*, 80:628–636, 2011.

[11] Future of radiation medicine symposium, Scottsdale, AZ, 2011. Report available at `http://www.elekta.com/future`.

[12] G Grigorov, T Kron, E Wong, J Chen, J Sollazzo, and G Rodrigues. Optimization of helical tomotherapy treatment plans for prostate cancer. *Phys. Med. Biol.*, 48(13):1933, 2003.

[13] H. W. Hamacher and M. Ehrgott. Editorial: Special section: Using discrete mathematics to model multileaf collimators in radiation therapy. *Discret. Appl. Math.*, 152(1-3):4–5, November 2005.

[14] P. Hansen and N. Mladenovic. An introduction to VNS. In S. Voss, S. Martello, I.H. Osman, and C. Roucairol, editors, *Meta- Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 223–233. Kluwer Academic Publishers, Boston, 1998.

[15] ICRU. Prescribing, reporting, and reporting photon beam therapy, 1993. International Commission on Radiation Units and Measurements, ICRU Report 50, Bethesda, Maryland, USA.

[16] H.M. Langer, V. Thai, and L. Papiez. Improved leaf sequencing reduces segments or monitor units needed to deliver IMRT using multileaf collimators. *J. Med. Phys.*, 28:12:1450–1458, 2001.

[17] T. S. Lawrence, R. K. Ten Haken, and A. Giaccia. Principles of radiation oncology. Cancer: Principles and Practice of Oncology. Lippincott Williams and Wilkins, 2008.

[18] L. Mason, V. Mak-Hau, and A. Ernst. An exact method for minimizing the total treatment time in intensity-modulated radiotherapy. *J. Oper. Res. Soc.*, 63:1447–1456, 2012.

[19] G. McCormick. Computability of global solutions to factorable nonconvex programs: Part i - convex underestimating problems. *Math. Program.*, 10:146–175, 1976.

[20] C. Men, X. Jia, S. Jiang, and H. E. Romeijn. Ultrafast treatment plan optimization for volumetric modulated arc therapy (VMAT). *J. Med. Phys.*, 37:5787–5791, 2010.

[21] N. Mladenovic and P. Hansen. Variable neighborhood search. *Comput. Oper. Res.*, 24:1097–1100, 1997.

[22] D. Nazareth, S. Brunner, M. Jones, H. Malhotra, and M. Bakhtiari. Optimization of beam angles for intensity modulated radiation therapy treatment planning using genetic algorithm on a distributed computing platform. *J. Med. Phys.*, 34:129–132, 2009.

[23] M. Oliver, M. Jensen, J. Chen, and E.Wong. Evaluation of optimization strategies and the effect of initial conditions on imat optimization using a leaf position optimization algorithm. *Phys. Med. Biol.*, 54:3543–3561, 2009.

[24] K. Otto. Volumetric modulated arc therapy: IMRT in a single gantry arc. *J. Med. Phys.*, 35:310–317, 2008.

[25] F. Peng, X. Jia, X. Gu, M.A. Epelman, H.E. Romeijn, and S.B Jiang. A new column-generation-based algorithm for VMAT treatment plan optimization. *Phys. Med. Biol.*, 57:4569–4588, 2012.

[26] H Edwin Romeijn, James F Dempsey, and Jonathan G Li. A unifying framework for multi-criteria fluence map optimization models. *Phys. Med. Biol.*, 49(10):1991, 2004.

[27] A. Schlaefer and A. Schweikard. Stepwise multi-criteria optimization for robotic radiosurgery. *J. Med. Phys.*, 35(5), 2008.

[28] M. Skórska and T. Piotrowski. Optimization of treatment planning parameters used in tomotherapy for prostate cancer patients. *Phys. Med.*, 29(3):273 – 285, 2013.

[29] Bofei Sun, Zhongshun Shi, Jie Song, Guangying Zhu, and Leyuan Shi. A linearized model and nested-partitions heuristics for vmat radiation treatment planning optimization. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 629–633, 2013.

[30] Bofei Sun, Jie Song, Guangying Zhu, and Leyuan Shi. A two-stage approach for VMAT treatment plan optimization. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 455–460, 2013.

[31] Z. C. Taskin, J. C. Smith, H. E. Romeijn, and J. F. Dempsey. Optimal multileaf collimator leaf sequencing in IMRT treatment planning. *Oper. Res.*, 58:675–690, 2010.

[32] Rapidarc brochure, 2007. Report available at `http://www.varian.com/media/oncology/solutions/pdf/RapidArc_brochure.pdf`. Last accessed on 03/11/2014.

[33] G. M. G. H. Wake. *Exact Minimisation of Treatment Time for the Delivery of Intensity Modulated Radiation Therapy*. PhD thesis, The University of Western Australia, 2009.

[34] C. Wang, S. Luan, G. Tang, D.Z. Chen, M.A. Earl, and C.X. Yu. Arc-modulated radiation therapy (amrt): a single-arc form of intensity-modulated arc therapy. *Phys. Med. Biol.*, 53:6291–6303, 2008.

[35] X. Wang, X. Zhang, L. Dong, H. Liu, Q. Wu, and R. Mohan. Development of methods for beam angle optimization for IMRT using an accelerated exhaustive search strategy. *Internat. J. Radiat. Oncol. Biol. Phys.*, 60:1325–1337, 2004.

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

[37] C.X. Yu and G. Tang. Intensity-modulated arc therapy: principles, technologies and clinical implementation. *Phys. Med. Biol.*, 56:R31–R54, 2011.

[38] M. Zeverino, S. Agostinelli, G Taccini, F. Cavagnetto, and S. Garelli. Advances in the implementation of helical tomotherapybased total marrow irradiation with a novel field junction technique. *Med. Dosim.*, 37:314–320, 2012.

[39] Hao Howard Zhang, Leyuan Shi, Robert Meyer, Daryl Nazareth, and Warren D'Souza. Solving beam-angle selection and dose optimization simultaneously via high-throughput computing. *INFORMS J. Comput.*, 21(3):427–444, 2009.

APPENDICES

The Formulation Adopted from the Model of [16]

In this section, we present a new MIP model for the treatment planning of VMAT based on an earlier model we presented in [4] and inspired by variables presented in [16] for intensity-modulated radiotherapy (IMRT). Similar to the model discussed in the paper, this new model also optimizes the MLC apertures and fluence weights simultaneously. Before we introduce the MIP model, we first present any new notation (in addition to the notation already defined in the paper), as follows:

- t_{ij}^k be a binary variable with $t_{ij}^k = 1$ if bixel i, j in snapshot k is open, and $t_{ij}^k = 0$ otherwise. [NB: this, and the next two variables are used in [16] for IMRT and in [29, 30] for VMAT];
- ℓ_{ij}^k be a binary variable with $\ell_{ij}^k = 1$ indicating the column j in row i of snapshot k is closed by the left leaf; and
- r_{ij}^k be a binary variable with $r_{ij}^k = 1$ indicating the column j in row i of snapshot k is closed by the right leaf.

First, we introduce the leaf-position constraints. The following set of constraints are presented in both [16] and [29, 30].

$$t_{ij}^k = 1 - \ell_{ij}^k - r_{ij}^k, \forall i \in I,\ j \in J,\ k \in K \tag{39}$$

$$r_{i,j}^k \leq r_{i,j+1}^k, \forall i \in I,\ j \in J \setminus \{1\} \tag{40}$$

$$\ell_{i,j}^k \geq \ell_{i,j+1}^k, \forall i \in I,\ j \in J \setminus \{1\} \tag{41}$$

Constraints (39)–(41) will ensure that the left and right leaves will not collide, the Consecutive-1 rule is observed, and that a bixel i, j will either be open, or be blocked by the left or the right leaf.

[29, 30] used the following two constraints to ensure consecutive snapshot leaf travel distance restrictions.

$$\ell_{i,j}^{k+1} \leq \ell_{i,j+\delta}^k, \forall i \in I,\ j = 1, \ldots, n - \delta,\ k = 1 \ldots |K| - 1 \tag{42}$$

$$r_{i,j}^{k+1} \geq \ell_{i,j-\delta}^k, \forall i \in I,\ j = \delta + 1, \ldots, n,\ k = 1 \ldots |K| - 1 \tag{43}$$

However, these constraints might produce infeasible solutions, as noted for a test problem instance available at http://www.deakin.edu.au/∼vicky/4-6-2-4-5-4-0.dat, where the MLC is discretized into 4 rows and 6 columns. The maximum leaf travel distance δ is 1. The optimal solution produced by the Sun *et al.* model has, e.g., $l^1_{12} = 1$, but $l^2_{11} = 0$, which results in a violation of the consecutive snapshot maximum leaf travel distance constraint.

We now describe how we obtain our consecutive snapshot leaf travel distance restriction constraints. Consider the left leaf in Row i of Snapshot k is in Position j, then $\ell^k_{i,j} = 1$ and $\ell^k_{i,j+1} = 0$, so $\ell^k_{i,j} - \ell^k_{i,j+1} > 0$. Now, in Snapshot $k+1$, the left leaf can only be in one of the following positions: $\{j-\delta, \ldots, j+\delta\}$. This means that exactly one of the differences of $(\ell^k_{i,j+\alpha} - \ell^k_{i,j+\alpha+1})$ will be > 0, for $\alpha = -\delta, \ldots, 0, \ldots, \delta$. Hence,

$$\sum_{\alpha=-\delta}^{\delta} (\ell^{k+1}_{i,j+\alpha} - \ell^{k+1}_{i,j+\alpha+1}) = \ell^{k+1}_{i,j-\delta} - \ell^{k+1}_{i,j+\delta+1} = 1$$

and therefore the consecutive snapshot leaf travel distance constraints for the left leaves are:

$$\begin{aligned}
\ell^k_{i,j} - \ell^k_{i,j+1} &\leq \ell^{k+1}_{i,j-\delta} - \ell^{k+1}_{i,j+\delta+1}, \\
\forall i \in I,\ j = \delta+1, \ldots, n-\delta-1,\ k = 1 \ldots |K|-1 \\
\ell^k_{i,j} - \ell^k_{i,j+1} &\leq \ell^{k+1}_{i,j-\delta}, \\
\forall i \in I,\ j = n-\delta, \ldots, n-1,\ k = 1 \ldots |K|-1 \\
\ell^k_{i,n} &\leq \ell^{k+1}_{i,n-\delta}, \\
\forall i \in I,\ k = 1 \ldots |K|-1 \\
\ell^k_{i,j} - \ell^k_{i,j+1} &\leq 1 - \ell^{k+1}_{i,j+\delta+1}, \\
\forall i \in I,\ j = 1, \ldots, \delta,\ k = 1 \ldots |K|-1 \\
\ell^{k+1}_{i,\delta+1} &\leq \sum_{j \in J} \ell^k_{i,j}, \\
\forall i \in I,\ k = 1 \ldots |K|-1
\end{aligned} \tag{44}$$

Similarly, the consecutive snapshot leaf travel distance constraints for the right leaves are:

$$
\begin{aligned}
r^k_{i,j} - r^k_{i,j-1} &\leq r^{k+1}_{i,j+\delta} - r^{k+1}_{i,j-\delta-1}, \\
\forall i \in I,\ j = \delta+2, \ldots, n-\delta,\ k = 1 \ldots |K|-1 \\
r^k_{i,j} - r^k_{i,j-1} &\leq 1 - r^{k+1}_{i,j-\delta-1}, \\
\forall i \in I,\ j = n-\delta+1, \ldots, n,\ k = 1 \ldots |K|-1 \\
r^k_{i,j} - r^k_{i,j-1} &\leq r^{k+1}_{i,j+\delta}, \\
\forall i \in I,\ j = 2, \ldots, \delta+1,\ k = 1 \ldots |K|-1 \\
r^k_{i,1} &\leq r^{k+1}_{i,1+\delta}, \\
\forall i \in I,\ k = 1 \ldots |K|-1 \\
r^{k+1}_{i,n-\delta} &\leq \sum_{j \in J} r^k_{i,j}, \\
\forall i \in I,\ k = 1 \ldots |K|-1 \qquad (45)
\end{aligned}
$$

The interleaf constraints are given by:

$$r^k_{i,j} + l^k_{i+1,j} \leq 1, \quad ,\forall j \in J,\ k \in K \qquad (46)$$

$$r^k_{i+1,j} + l^k_{i,j} \leq 1, \quad ,\forall j \in J,\ k \in K \qquad (47)$$

These are presented in [16].

We adopt the dose constraints from our model presented in the paper.

$$z^k - z^{k+1} \leq \Delta \qquad \forall k \in K \qquad (48)$$

$$z^{k+1} - z^k \leq \Delta \qquad \forall k \in K \qquad (49)$$

$$d_v \geq L_v \qquad v \in V_t \qquad (50)$$

$$d_v \leq U_v \qquad v \in V \qquad (51)$$

$$d_v \geq \bar{d} x_v \qquad v \in V_t \qquad (52)$$

$$x \in \{0,1\}^{|V_t|}; 0 \leq z^k \leq \bar{M}^{|K|} \qquad (53)$$

Also adopted from the model in the paper is the dose calculation equation:

$$d_v = \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \bar{z}^k_{ij} \times D^k_{ijv} \qquad (54)$$

As for the linearization constraints, we adopt Constraints (15)–(18) of the model in the paper, replacing $\sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} y^k_{i(\ell,r)}$ with t^k_{ij}. Hence we have, for each $k \in K, i \in I, j \in J$:

$$\bar{z}^k_{ij} \leq \bar{M} t^k_{ij} \tag{55}$$

$$\bar{z}^k_{ij} \leq z^k \tag{56}$$

$$\bar{z}^k_{ij} \geq \bar{M}\left(-1 + t^k_{ij}\right) + z^k \tag{57}$$

$$\bar{z}^k_{ij} \geq 0 \tag{58}$$

For the objective function, as in the model presented in the paper, we have different options and there is no new notation to introduce here. This concludes the presentation of this formulation.

Detailed Technical Results

Proof. **PROOF OF PROPOSITION 5.** Here we prove the trivial facets of $P_{1\times n\times|K|\times 1}$.

(a) This is straightforward, as one can simply remove the last point from the proof of Proposition 4 and obtain sufficient number of affinely independent points with $x_v = 0$.

(b) This is also straightforward, as one can simply re-use all cases used in the proof of Proposition 4 (except for Case (1b)) with the following minor modifications to satisfy $x_v = 1$:

  (i) Replace L with U in all instances;

  (ii) In Case (1c), $L + \epsilon$ is replaced with $U - \epsilon$, for some ϵ such that $\epsilon < U - \bar{d}$; and

  (iii) $x_v = 0$ is replaced with $x_v = 1$ in all cases.

(c) Here we have two cases: (i) $y^k_{(\ell,r)}$ for all $k \in K$ and $r \neq \ell + 2$; and (ii) $y^k_{(\ell,r)}$ for all $k \in K$ and $r = \ell + 2$. Let $\hat{k}, \bar{k} \in K$ such that $\hat{k} \neq \bar{k}$.

  (i) We can use all the vectors constructed in the proof of Proposition 4, except for the vector with $y^k_{(\ell,r)} = 1$, and for the cases where $r = \ell + 1$, $(\tau, \tau + 1)$ will be arbitrarily chosen from $\{(h, h+1) \mid h = 0, \ldots, n\} \setminus \{(\ell, r)\}$.

  (ii) From the vectors constructed in the proof of Proposition 4, for all $y^k_{(j-1,j+1)}$ where $k \in K$ and $j \in \{0, \ldots, n\}$, there are exactly two vectors with $y^k_{(j-1,j+1)} = 1$. We

remove both of these vectors, use all of the rest of the vectors, and then add the following vector.

$y^k_{(j-1,j+2)} = 1$, when $j \leq n-1$, $z^k = \bar{z}^k_j = \bar{z}^k_{j+1} = \frac{L}{nD^k_{(j,j+1)}}$; (or, $y^k_{(j-2,j+1)} = 1$, when $j = n$, $z^k = \bar{z}^k_{j-1} = \bar{z}^k_j = \frac{L}{nD^k_{(j-1,j)}}$;) $y^\beta_{(0,n+1)} = 1$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise; $\bar{z}^\beta_{j'} = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for all $j' = 1, \ldots, n$; $z^\beta = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise; $y^{k'}_{(\tau,\tau+1)} = 1$, for all $k' \in K \setminus \{k, \beta\}$ for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise, and all other variables equal zero. (Note this implies that an additional condition is given by $|K| = 4$.)

(d) This is very similar to the proof of Case c(i). We reuse all the affinely independent vectors used in the proof of Proposition 4, except the vector with $z^k_j = 0$, with $\hat{k} \neq k$ and $\bar{k} \neq k$.

(e) In all the affinely independent vectors used in the proof of Proposition 4, there are precisely n vectors where $\bar{z}^k_j - z^k \neq 0$. One of such case is in Case 1b in the proof of Proposition 4, and we can simply remove this point, as the dimension of the facet is one less than that of the polytope. The rest of the $n-1$ vectors where $\bar{z}^k_j - z^k \neq 0$ appear in Case 3 of the proof of Proposition 4, when $\alpha \in \{1, \ldots, n\} \setminus \{j\}$. These vectors are as follows:

$y^k_{(\alpha-1,\alpha+1)} = 1$, $z^k = \bar{z}^k_\alpha = \frac{L}{nD^k_\alpha}$, (whilst $\bar{z}^k_j = 0$, hence $z^k \neq \bar{z}^k_j$); $y^\beta_{(0,n+1)} = 1$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise; $\bar{z}^\beta_{j'} = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for all $j' = 1, \ldots, n$; $z^\beta = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise; $y^{k'}_{(\tau,\tau+1)} = 1$, for all $k' \in K \setminus \{k, \beta\}$ for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise, and all other variables equal zero.

We remove all these vectors, and replace them with the following $(n-1)$ vectors, where we set $y^k_{(j-1,j+\gamma)}$ variables for $\gamma = 2, \ldots, n+1-j$, (there are $n-j$ of them for each k), and $y^k_{(j+\sigma,j+1)}$ variables for $\sigma = 0, \ldots, j-2$ (there are $j-1$ of them for each k) to a value of 1 instead. We can define these vectors formally as below:

$y^k_{(j-1,j+\gamma)} = 1$, $z^k = \bar{z}^k_j = \ldots = \bar{z}^k_{j+\gamma-1} = \frac{L}{nD^k_{(j,j+\gamma-1)}}$; (or, $y^k_{(j-\sigma,j+1)} = 1$, $z^k = \bar{z}^k_{j-\sigma+1} = \ldots = \bar{z}^k_j = \frac{L}{nD^k_{(j-\sigma+1,j)}}$); $y^\beta_{(0,n+1)} = 1$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise; $\bar{z}^\beta_{j'} = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for all $j' = 1, \ldots, n$; $z^\beta = \frac{(n-1)L}{nD^\beta_{(1,n)}}$, for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise; $y^{k'}_{(\tau,\tau+1)} = 1$, for all $k' \in K \setminus \{k, \beta\}$ for $\beta = \bar{k}$ if $k = \hat{k}$, and $\beta = \hat{k}$ otherwise, and all other variables equal zero. □

Proof. **PROOF OF PROPOSITION 7.** First of all, note that due to constraint (2), we have:

$$\sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell\geq j \bigvee r\leq j}} y_{\ell,r} = 1 - \sum_{\substack{(\ell,r)\in\mathcal{L}\\ \ell<j<r}} y_{\ell,r}$$

If for any given $(\ell, r) \in \mathcal{L}$ such that $\ell \geq j$ or $r \leq j$, $y_{\ell,r} = 1$, then $\bar{z}_j = 0$, and $z \geq \frac{L}{D_{\ell+1,r-1}}$ has to be satisfied due to (9). On the other hand, if for any given $(\ell, r) \in \mathcal{L}$ such that $\ell < j < r$, $y_{\ell,r} = 1$, then (24) reduces to $\bar{z}_j \leq z$. Therefore, (24) is valid. The dominance of (24) over (16) follows simply that $L \geq 0$.

To prove its facet-defining property, first let $S_j = \{(\ell, r) \in \mathcal{L} | \ell \leq j \leq r\}$. Also let $d' = \min\{\bar{d}, \bar{M}D_j\}$. Consider the following $\frac{n(n+1)}{2} + n + 1$ affinely independent points:

- $\frac{n(n+1)}{2}$ points, where $y_{\ell r} = 1$ holds for one (ℓ, r), $\bar{z}_{\widehat{j}} = L/D_{\ell+1,r-1}$ where $\ell < \widehat{j} < r$, $z = L/D_{\ell+1,r-1}$, and all other variables zero.

- n points, where $y_{\ell r} = 1$ holds for one $(\ell, r) \in S_j$, all $\bar{z}_{\widehat{j}} = d'/D_{\ell+1,r-1}$ where $\ell < \widehat{j} < r$, $z = d'/D_{\ell+1,r-1}$, and all other variables zero.

- 1 point, where $y_{0n+1} = 1$, $\bar{z}_j = \bar{d}/D_{1,n}$, $z = \bar{d}/D_{1,n}$, $x = 1$, and all other variables zero. □

Proof. **PROOF OF PROPOSITION 8.** First, note that due to constraint (2), we have one of the following possibilities:

(a) $y_{\ell,r} = 1$ for a given $(\ell, r) \in \mathcal{L}$ such that $\ell < j$ and $\widehat{j} < r$.

(b) $y_{\ell,r} = 1$ for a given $(\ell, r) \in \mathcal{L}$ such that either $\ell \geq \widehat{j}$ or $r \leq j$ or $\ell \geq j$ and $r \leq \widehat{j}$.

(c) $y_{\ell,r} = 1$ for a given $(\ell, r) \in \mathcal{L}$ such that either $\ell < j$ and $j < r \leq \widehat{j}$ or $\widehat{j} > \ell \geq j$ and $r > \widehat{j}$.

If (a) is the case, then we know that $\bar{z}_j = \bar{z}_{\widehat{j}} = z$ and hence the left-hand side of both inequalities reduce to z, whereas the right-hand side simply reduces to $\min\{\bar{M}, \frac{U}{D_{\ell+1,r-1}}\}$ in (25) and to $\frac{L}{D_{\ell+1,r-1}}$ in (26). The first condition holds due to (10) and $z \leq \bar{M}$, and the second condition holds due to (9).

If (b) is the case, then we know that $\bar{z}_j = 0 = \bar{z}_{\widehat{j}}$ and hence the left-hand side of both inequalities reduce to $-z$, whereas the right-hand side simply reduces to $\frac{L}{D_{\ell+1,r-1}}$ in (25) and

to $\min\{\bar{M}, \frac{U}{D_{\ell+1,r-1}}\}$ in (26). The first condition holds due to (9), and the second condition holds due to (10) and $z \leq \bar{M}$.

If (c) is the case, then right-hand side of both constraints becomes simply 0, and either $\bar{z}_j = 0$ or $\bar{z}_{\widehat{j}} = 0$ holds. This concludes the validity proof.

To prove facet-defining property of (25), first let $S_j = \{(\ell, r) \in \mathcal{L} | (\ell < j \bigwedge j < r \leq \widehat{j}) \bigvee (\widehat{j} > \ell \geq j \bigwedge r > \widehat{j})\}$, i.e., the case (c). Consider the following $\frac{n(n+1)}{2} + n + 1$ affinely independent points:

- $\frac{n(n+1)}{2}$ points, where $y_{\ell r} = 1$ holds for one (ℓ, r), and
  - if $\ell < j < \widehat{j} < r$, then $\bar{z}_{j''} = \min\{\bar{M}, U/D_{\ell+1,r-1}\}$ where $\ell < j'' < r$, $z = \min\{\bar{M}, U/D_{\ell+1,r-1}\}$, and all other variables zero.
  - else $\bar{z}_{j''} = L/D_{\ell+1,r-1}$ where $\ell < j'' < r$, $z = L/D_{\ell+1,r-1}$, and all other variables zero.
- n points, where $y_{\ell r} = 1$ holds for one $(\ell, r) \in S_j$, all $\bar{z}_{j''} = d'/D_{\ell+1,r-1}$ where $\ell < j'' < r$ and $d' = \min\{\bar{d}, \min_{\ell < j'' < r} \bar{M} D_{j''}\}$, $z = d'/D_{\ell+1,r-1}$, and all other variables zero.
- 1 point, where $y_{0,n+1} = 1$, $\bar{z}_{j''} = U/D_{1,n}$, $z = U/D_{1,n}$, $x = 1$, and all other variables zero.

To prove facet-defining property of (26), first let $S_j = \{(\ell, r) \in \mathcal{L} | (\ell < j \bigwedge j < r \leq \widehat{j}) \bigvee (\widehat{j} > \ell \geq j \bigwedge r > \widehat{j})\}$, i.e., the case (c). Consider the following $\frac{n(n+1)}{2} + n + 1$ affinely independent points:

- $\frac{n(n+1)}{2}$ points, where $y_{\ell r} = 1$ holds for one (ℓ, r), and
  - if $\ell < j < \widehat{j} < r$, then $\bar{z}_{j''} = L/D_{\ell+1,r-1}$ where $\ell < j'' < r$, $z = L/D_{\ell+1,r-1}$, and all other variables zero.
  - else $\bar{z}_{j''} = \min\{\bar{M}, U/D_{\ell+1,r-1}\}$ where $\ell < j'' < r$, $z = \min\{\bar{M}, U/D_{\ell+1,r-1}\}$, and all other variables zero.
- n points, where $y_{\ell r} = 1$ holds for one $(\ell, r) \in S_j$, all $\bar{z}_{j''} = d'/D_{\ell+1,r-1}$ where $\ell < j'' < r$ and $d' = \min\{\bar{d}, \min_{\ell < j'' < r} \bar{M} D_{j''}\}$, $z = d'/D_{\ell+1,r-1}$, and all other variables zero.
- 1 point, where $y_{\ell,r} = 1$ for (ℓ, r) such that $\ell \geq j$ or $\widehat{j} \geq r$ and $\bar{M} D_{\ell+1,r-1} \geq \bar{d}$; $\bar{z}_{j''} = U/D_{\ell+1,r-1}$, $z = U/D_{\ell+1,r-1}$, $x = 1$, and all other variables zero. □

Detailed Numerical Results

Table 9 Lower/upper bounds after 1 hour default Cplex run, with $cplex.epgap = 0.01$. Time/Gap column indicates time if < 3600 sec., otherwise $gap = \frac{(UB-LB)}{LB}$. No upper bounds could be found for any other instances with MLC size of 15×15 or above.

Problem Instance	LB	UB	Time/Gap
6-6-6-6-6/16-0.1-5-2-2-3	17	86	405.88%
7-7-7-7-7/8-0.08-5-1-1-3	273	274	801
7-7-7-7-7/8-0.1-5-1-1-3	103	118	14.56%
7-7-7-7-7/8-0.2-5-1-1-3	6	33	450.00%
7-7-7-7-7/12-0.08-5-1-1-3	NA	69	∞
7-7-7-7-7/12-0.1-5-1-1-3	NA	154	∞
7-7-7-7-7/12-0.2-5-1-1-3	0	92	∞
7-7-7-7-7/16-0.08-5-1-1-3	NA	160	∞
7-7-7-7-7/16-0.1-5-1-1-3	NA	73	∞
7-7-7-7-7/16-0.2-5-1-1-3	NA	38	∞
8-8-8-8-8/16-0.1-5-2-2-3	27	426	1477%
10-10-10-10-10/32-0.3-10-3-3-5	NA	995	∞
12-12-12-12-12/8-0.1-5-2-3-3	NA	1611	∞
12-12-12-12-12/16-0.1-3-2-3-2	1728	1728	993
12-12-12-12-12/16-0.08-5-2-3-3	NA	1719	∞
12-12-12-12-12/16-0.1-8-2-3-5	NA	1642	∞
12-12-12-12-12/16-0.1-5-1-2-3	NA	1628	∞
12-12-12-12-12/16-0.1-5-2-3-3	NA	1609	∞
12-12-12-12-12/16-0.1-5-3-4-3	312	1710	448.08%
12-12-12-12-12/16-0.2-5-2-3-3	NA	1649	∞
12-12-12-12-12/24-0.1-5-2-3-3	NA	1728	∞
15-15-15-15-15/8-0.1-5-2-3-3	NA	3362	∞

Table 10 LRHeur1 results for medium (MLC size 12×12) and large (MLC size 15×15) problems, with *cplex.tilim* = 900 for subroutines and *cplex.tilim* = 1200 for fixed heuristic. * indicate instances where Cplex stopped due to no memory left (hence total number of iterations $< |K|$).

	# iterations				Objective values		
Problem Instance	1st sol	Best sol	Total	# sol's	1st sol	Best sol	Time
12-12-12-12-12/8-0.1-5-2-3-3	1	2	8	8	149	235	5h24m
12-12-12-12-12/16-0.1-3-2-3-2	1	1	16	16	1728	1728	1h13m
12-12-12-12-12/16-0.08-5-2-3-3	1	2	9*	8	1155	1344	5h56m
12-12-12-12-12/16-0.1-8-2-3-5	1	4	16	16	343	500	11h02m
12-12-12-12-12/16-0.1-5-1-2-3	1	1	16	16	350	350	14h12m
12-12-12-12-12/16-0.1-5-2-3-3	1	8	16	16	207	292	14h37m
12-12-12-12-12/16-0.1-5-3-4-3	1	3	16	16	714	768	10h26m
12-12-12-12-12/16-0.2-5-2-3-3	1	1	16	16	566	566	11h01m
12-12-12-12-12/24-0.1-5-2-3-3	1	1	24	24	1728	1728	9h43m
15-15-15-15-15/8-0.1-5-2-3-3	1	4	8	8	1891	2213	5h37m
15-15-15-15-15/16-0.1-8-2-3-5	1	4	16	16	2188	2928	15h49m
15-15-15-15-15/16-0.08-5-2-3-3	1	2	5*	4	3324	3352	3h28m
15-15-15-15-15/16-0.1-3-2-3-2	1	1	16	16	3375	3375	1h13m
15-15-15-15-15/16-0.1-5-1-2-3	1	3	5*	4	3103	3175	3h38m
15-15-15-15-15/16-0.1-5-2-3-3	1	3	16	16	3374	3375	2h43m
15-15-15-15-15/16-0.1-5-3-5-3	1	1	16	16	3375	3375	2h46m
15-15-15-15-15/16-0.2-5-2-3-3	1	5	16	16	3093	3153	12h14m
15-15-15-15-15/24-0.1-5-2-3-3	1	1	24	23	3375	3375	5h05m

Table 11 LRHeur2 results for medium (MLC size 12×12) problems, with *cplex.tilim* = 900 for subroutines and *cplex.tilim* = 1200 for fixed heuristic. No feasible solution for any of the large (MLC size 15×15) problems could be found after 20 iterations. * indicate instances where Cplex stopped due to no memory left (hence total number of iterations < 20).

	# iterations				Objective values		
Problem Instance	1st sol	Best sol	Total	# sol's	1st sol	Best sol	Time
12-12-12-12-12/8-0.1-5-2-3-3	1	4	20	19	163	291	13h40m
12-12-12-12-12/16-0.1-3-2-3-2	1	1	20	16	1728	1728	1h26m
12-12-12-12-12/16-0.08-5-2-3-3	1	6	20	18	1118	1222	13h39m
12-12-12-12-12/16-0.1-8-2-3-5	1	1	2*	1	381	381	1h1m
12-12-12-12-12/16-0.1-5-1-2-3	2	3	5*	3	59	227	2h55m
12-12-12-12-12/16-0.1-5-2-3-3	1	1	2*	1	406	406	1h1m
12-12-12-12-12/16-0.1-5-3-4-3	1	1	19*	16	811	811	12h25m
12-12-12-12-12/16-0.2-5-2-3-3	1	1	2*	1	386	386	3599
12-12-12-12-12/24-0.1-5-2-3-3	1	1	10*	6	1727	1727	4h38m

Table 12 CentHeur results for medium (MLC size 12×12) and large (MLC size 15×15) problems, with *cplex.tilim* = 900 for subroutines and *cplex.tilim* = 1200 for fixed heuristic. * indicate instances where Cplex stopped due to no memory left (hence total number of iterations $< |K|$).

	# iterations				Objective values		
Problem Instance	1st sol	Best sol	Total	# sol's	1st sol	Best sol	Time
12-12-12-12-12/8-0.1-5-2-3-3	2	8	8	7	65	199	3h50m
12-12-12-12-12/16-0.1-3-2-3-2	3	8	16	14	1084	1728	1h22m
12-12-12-12-12/16-0.08-5-2-3-3	3	14	16	14	244	1330	8h47m
12-12-12-12-12/16-0.1-8-2-3-5	3	16	16	14	18	419	8h47m
12-12-12-12-12/16-0.1-5-1-2-3	2	14	16	15	98	259	15h06m
12-12-12-12-12/16-0.1-5-2-3-3	2	13	16	12	46	162	23h05m
12-12-12-12-12/16-0.1-5-3-4-3	2	15	16	15	36	846	9h
12-12-12-12-12/16-0.2-5-2-3-3	2	12	16	15	71	314	9h29m
12-12-12-12-12/24-0.1-5-2-3-3	3	21	24	21	585	1728	9h
15-15-15-15-15/8-0.1-5-2-3-3	2	6	7*	5	84	1615	2h31m
15-15-15-15-15/16-0.1-8-2-3-5	2	6	7*	5	182	1360	11h05m
15-15-15-15-15/16-0.08-5-2-3-3	2	9	10*	8	1287	3042	5h08m
15-15-15-15-15/16-0.1-3-2-3-2	2	11	16	15	2169	3375	1h14m
15-15-15-15-15/16-0.1-5-1-2-3	2	12	14*	12	1212	2870	7h47m
15-15-15-15-15/16-0.1-5-2-3-3	2	3	4	2	695	1788	3355
15-15-15-15-15/16-0.1-5-3-5-3	3	11	16	14	203	3375	3h36m
15-15-15-15-15/16-0.2-5-2-3-3	2	16	16	15	733	3158	9h41m
15-15-15-15-15/24-0.1-5-2-3-3	4	7	24	21	3261	3375	1h06m