Fast Simultaneous Angle, Wedge, and Beam Intensity Optimization in Inverse Radiotherapy Planning

Konrad Engel
Universität Rostock
Fachbereich Mathematik
18051 Rostock
Germany

Eckhard Tabbert
Radiologische Klinik Schwerin
Lübecker Str. 276
D–19049 Schwerin
Germany

Abstract

We present a new fast radiotherapy planning algorithm which determines approximatively optimal gantry and table angles, kinds of wedges, leaf positions and intensities simultaneously in a global way. Other parameters are optimized only independently of each other. The algorithm uses an elaborate field management and field reduction. Beam intensities are determined via a variant of a projected Newton method of Bertsekas. The objective function is a standard piecewise quadratic penalty function, but it is built with efficient upper bounds which are calculated during the optimization process. Instead of pencil beams, basic leaf positions are included. The algorithm is implemented in the new beam modelling and dose optimization module Homo OptiS.

1 Introduction

"Inverse radiation therapy planning" is a well–known notion for techniques providing the geometric–physical set–up and the intensity profiles of radiation beams that realize a desired dose distribution for a particular patient.

Here the geometric–physical set–up is given by a certain number of fields having parameters like position of the isocenter, field–width, field–length, gantry angle, table angle, collimator angle, kind of wedge, shape of a compensator, and energy. The intensity profile is characterized either only by the time of radiation (old version) or by the time of radiation of a number of pencil beams (beamlets, bixels) into which a beam is partitioned. These intensity modulated beams are realized by multileaf–collimators (MLC). For an overview cf. [12, 23, 54, 56].

The clinical requirements to the dose distribution can more or less never be realized, but by means of optimization the deviation from the requirements can be kept "small". Nowadays, the optimization is an iterated two–stage process. In the first stage the geometric–physical set–up is fixed and in the second stage the intensity profiles are computed. The first stage is carried out by experienced trial and error, by heuristic (often geometrically based) methods, or by stochastic search methods including in particular simulated annealing [9, 36, 39, 42, 47, 53] and genetic algorithms [21, 23, 31]. These search methods are also time–consuming in their fast variants. For the second stage one uses algorithms from linear programming [2, 6, 25, 46] or nonlinear, in particular quadratic, programming [12, 14, 15, 16, 18, 26, 29, 37, 45, 48, 50, 51, 52, 59] and control theory [28] including multiple objective approaches [17, 24, 60]. Also iterative dose reconstruction techniques [8, 27, 43, 49] and methods from global optimization have been developped [58]. Moreover, it must be mentioned that one–stage algorithms have been designed by means of mixed integer programming [13, 33, 34] but they are time– and memory–consuming, too.

We present a new method that realizes on the one hand a near–optimal geometric–physical set–up for the most important parameters gantry angle, table angle, kind of wedge and energy (optional) and on the other hand optimal beam intensities in a **one–stage process**. Other parameters are determined automatically by efficient heuristics. We emphasize that it is not necessary to fix the geometric–physical set–up in advance. This method is already realized in our beam modelling and dose optimization module HOMO OPTIS which computes the solution – depending on the concrete situation – in a few seconds or at most in a few minutes.

Essential new ingredients of the algorithm are the **determination of voxel–dependent bounds and costs**, a **special variant of a projected**

Newton method, a well–devised **field management and a field reduction** as well as the use of **basic leaf positions** instead of pencil beams.

Because of the large number of possible fields, a huge number of dose calculations are necessary. We work with high energy photon beams and use a fast 3D–ray tracing algorithm mETMR (modified equivalent tissue–maximum–ratio) with radiological depth correction and a modified scatter model for field size and tissue–inhomogeneity scatter effects. Thus, in particular heterogeneity is considered in the dose calculation [5, 11, 19, 38, 41]. The time–consuming kernel methods and Monte–Carlo–methods more precisely take into account lateral scattering effects caused by the density and hence electron disequilibrium. These effects occur especially on tissue boundaries and in the case of a small field size and high energies [1, 10, 35, 55]. We carried out computations where perturbations in dose calculations were included, i.e. we simulated deviations from the correct dose values. It turned out that these perturbations do not significantly influence the quality of the solution which shows that our dose calculation algorithm is sufficient for the optimization. The robustness of the method can be explained by the choice of voxel–dependent bounds determined by dose calculations and by averaging effects because a treatment plan does not consist of one field only, but of several fields.

The size of the voxels used in the program are CT-slice dependent. Depending on the patient and on the location the size is between $3 \times 3 \times 5$ mm^3 and $7 \times 7 \times 10$ mm^3. Smaller size increases the number of voxels but does not yield significantly better results.

We used a standard piecewise quadratic objective function. Clearly, one is interested in a high tumor control probability (TCP) and in a small normal tissue complication probability (NTCP). But, as emphasized by several authors, [28] at the moment there does not exist a commonly accepted model for calculating these probabilities. At least heuristically it seems to be clear that an optimal value of the piecewise quadratic objective function corresponds to optimal values of the aforementioned probabilities. Moreover, quadratic functions can be handled very well numerically.

Our new method accelerates and improves the radiotherapy treatment planning. In order to explain the ideas and their effects we cannot avoid discussing the optimization in mathematical detail.

2 Notation and terminology

An elaborate field management plays an essential role in our approach. Since it must be implemented in an object–oriented way we use for several important parameters a C++–similar notation. Let a *field* F be a class having the following parameters:

(1) $F \to I$ (position of the isocenter)
(2) $F \to W$ (field width)
(3) $F \to L$ (field length)
(4) $F \to C$ (the positions of the leaves of an MLC)
(5) $F \to E$ (energy)
(6) $F \to \beta$ (field angle, i.e. collimator rotation)
(7) $F \to \varphi$ (gantry angle)
(8) $F \to \theta$ (table angle)
(9) $F \to \kappa$ (kind of wedge)

A 10th distinguished parameter of a field F is its weight x_F (i.e. the time) which we consider separately. A *treatment plan* is a pair $(\mathcal{F}, x)$ where $\mathcal{F}$ is a (small) set of fields and x is a function $\mathcal{F} \to \mathbb{R}_+$ representing the field weights. We write briefly x_F instead of $x(F)$ for all $F \in \mathcal{F}$.

We emphasize that intensity modulated fields are usually realized as a superposition of fields which differ only by the position of the leaves and by the weight (step and shoot). Therefore we consider intensity modulated fields as certain sets of fields.

Moreover, let a *voxel* v be a class having the following parameters:

(1) $v \to P$ (position and size in a given coordinate system)
(2) $v \to D$ (Electron density of the voxel)
(3) $v \to R$ (kind of region)

The (abstract) *patient* is a finite set V of voxels. The kind of region yields a partition of the patient V into blocks. Usually, we have one block T which is called the *planning target volume* (PTV), a family $\mathcal{R}$ of blocks R which are the *organs at risk* (OAR), and one block S which consists of the remaining voxels of V, i.e. the rest of body (ROB),

$$V = \left(\bigcup_{R \in \mathcal{R}} R \right) \cup S \cup T.$$

If there are more than one target, the set T must be replaced everywhere by a union of the form $\cup_{T\in\mathcal{T}}T$. We emphasize that we do not allow intersections of regions of interest.

In practice, these regions are given by contours drawn by the physician or physicist on each CT–slice. In Figure 1 there is given one CT–slice containing contours of the PTV and the OARs. This slice is illustrated on the left by means of the Hounsfield–values and on the right by means of the Electron density values (for dose calculation) in a low resolution which is sufficient for the optimization. The voxels (on the right) are represented by (two–dimensional) squares.

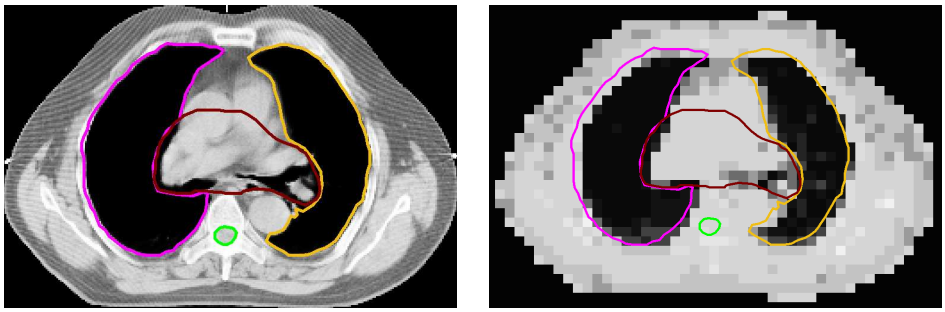

Figure 1: CT–slices representing Hounsfield values (left) and Electron density values of voxels (right)

In this paper we suppose that there is given a fast algorithm which determines for each field F and each voxel v the dose $D_F(v)$ which is accumulated at v by the field F with unitary weight $x_F = 1$. It is well–known that we have in the non–unitary case

$$D_{(F,x)}(v) = x_F D_F(v)$$

and that the total dose $D_{(\mathcal{F},x)}(v)$ which is accumulated at v by the treatment plan $(\mathcal{F}, x)$ is given by

$$D_{(\mathcal{F},x)}(v) = \sum_{F\in\mathcal{F}} x_F D_F(v).$$

3 Objective function

The physician fixes a value b_T, i.e. the prescribed dose to the target, and the aim is to find a treatment plan $(\mathcal{F}, x)$ such that $D_{(\mathcal{F},x)}(v)$ is very near to b_T for each $v \in T$ and that $D_{(\mathcal{F},x)}(v)$ is as small as possible for each other voxel, but in particular for the voxels in the OARs. It is a conventional method to fix a value b_R for each $R \in \mathcal{R}$ and to write the conditions

$$\begin{aligned} D_{(\mathcal{F},x)}(v) &= b_T \text{ for all } v \in T \\ D_{(\mathcal{F},x)}(v) &\leq b_R \text{ for all } v \in R, R \in \mathcal{R}. \end{aligned}$$

This model has two disadvantages. Firstly, the choice of b_R has a subjective flavor and intuition as well as experience are necessary. Secondly, for voxels which are far away from the PTV, the bound is often satisfied automatically whereas for voxels near to the PTV the bound cannot be kept. Thus we propose to fix for each voxel $v \in V \setminus T$ an individual bound b_v in order to have finally

$$\begin{aligned} D_{(\mathcal{F},x)}(v) &= b_T \text{ for all } v \in T \\ D_{(\mathcal{F},x)}(v) &\leq b_v \text{ for all } v \in V \setminus T. \end{aligned}$$

In Section 5 we describe an algorithm for the calculation of the bounds b_v.

Assume for a moment that $\mathcal{F}$ is fixed. Using the vector notation

$$\boldsymbol{d}_v = (D_F(v))_{F \in \mathcal{F}}, \quad \boldsymbol{x} = (x_F)_{F \in \mathcal{F}}$$

the system above can be rewritten as

$$\begin{aligned} \boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} &= b_T \text{ for all } v \in T \\ \boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} &\leq b_v \text{ for all } v \in V \setminus T \\ \boldsymbol{x} &\geq \boldsymbol{0}. \end{aligned}$$

In almost all practical situations this system does not have an admissible solution. Thus we must search for "almost" admissible solutions. As mentioned in the introduction, there are standard linear and quadratic programming techniques for doing this. To achieve high speed algorithms we prefer quadratic programming. Let the real function z_+ be defined by

$$z_+ = \begin{cases} z, & \text{if } z \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Associating with each voxel v an *importance factor* c_v, we come to the following problem where a *penalty function* $f(\boldsymbol{x})$ must be minimized subject to the nonnegativity condition:

$$f(\boldsymbol{x}) \to \min$$
$$\boldsymbol{x} \geq \mathbf{0}$$

where

$$f(\boldsymbol{x}) = \sum_{v \in T} c_v (\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_T)^2 + \sum_{R \in \mathcal{R}} \sum_{v \in R} c_v (\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_v)_+^2 + \sum_{v \in S} c_v (\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_v)_+^2. \quad (1)$$

We call this problem the *weight optimization problem* (WOP). Hints for the choice of the importance factors are given in Section 4. In Section 6 we describe a fast algorithm for the WOP. In (1), for every target voxel a deviation from b_T is penalized. In a modified model, one can generate similarly a penalty if $\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} < b_T$ or $\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} > \overline{b}_T$, where b_T and $\overline{b}_T$ are lower and upper bounds for the prescribed dose, respectively.

The more difficult problem is the determination of the set of fields $\mathcal{F}$ which gives minimal penalty values. We propose to fix the field parameters (1)–(6) independently of the other fields in an "optimal way", i.e. to do a **one–field–optimization** for some properties. This is explained in Section 7. The main result of the paper is an approximation algorithm for an optimal choice of the field parameters (7)–(9), i.e. angles and wedges are **optimized globally**. This can be achieved by combining the fast algorithm for the WOP with a sophisticated *field management* which is presented in Section 8. If the number of fields is too large, a well devised *field reduction*, explained in Section 9, provides the final treatment plan. In order to speed up the computations and to decrease memory demand, we do not really work with all voxels. We choose voxels randomly and forget all other voxels during the optimization process. The random choice is described in Section 10.

First we place the leaves of an MLC in such a way that no part of the PTV is covered by the leaves in the beam's eye view. This seems to be sufficient for convex PTVs. In the non–convex case, a partial covering of the PTV should be allowed, therefore the first placement will be modified in several ways. The optimal placement of the leaves in this case will be discussed in Sections 11 and 12 where the *leaf–setting* is included in a specific way in the field management.

4 Importance factors

Though there exist algorithms for the determination of the importance factor, [57], we believe that the choice of the importance factors in the objective function (1) of the WOP is, at the moment, not a mathematical problem. The physician must decide how important the conditions on the regions are. Increasing the factor c_v for one voxel v improves the results for this voxel but makes, in general, results for other voxels worse. Thus a carefully directed, dose–volume histogram adequate variation of the importance factors seems to be unavoidable. But in order to be relatively independent of the concrete structure of the patient we have the following propositions:

First choose for each region $T, R \in \mathcal{R}, S$ a *general importance factor* c_T, c_R, c_S (i.e. do variations only for these numbers). Then put for $v \in R \in \mathcal{R}$ (and analogously for T and S)

$$c_v = \frac{c_R}{\text{size of } R}.$$

Here *size of* R could be e.g. the sum of the volumes of the voxels being contained in R. Such a normalization was also used before, [28]. It is heuristically clear (and experiments confirm this) that voxels at the boundary of the regions are especially important, in particular boundary target voxels. Low-dose values for some of these voxels make the treatment plan already unacceptable for the physician.

Thus we propose to multiply c_v with a factor greater than 1, say 2, for these voxels. Finally, we make a small anticipation to the next section where we calculate for each non–target voxel v an efficient upper bound b_v. If b_v is "almost" zero, then the voxel v seems to be relatively unimportant whereas voxels v with "large" b_v seem to be important. Thus, having already calculated b_v, we put (with $b_v = b_T$ for each target voxel)

$$c_v := c_v(b_v + C),$$

where C is some constant which is proportional to b_T, e.g. $C = b_T/4$.

5 Upper bounds

Let $\mathcal{F}$ be a large set of, say 360, fields which can potentially serve as elements of a practical treatment plan. We consider the treatment in an "aggressive

manner", i.e. our first aim is to avoid target voxels for which the dose is less than the desired value. Let $v \in V \setminus T$ be a non–target voxel. The minimal dose–value for v, using the aggressive method, is obviously the minimum value of the objective function in the following linear programming problem (LPP):

$$\begin{aligned} \boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{x} &\to \min \\ \boldsymbol{d}_w^{\mathrm{T}}\boldsymbol{x} &\geq b_T \text{ for all } w \in T \\ \boldsymbol{x} &\geq \boldsymbol{0}. \end{aligned}$$

We could put the upper bound b_v equal to this minimum value. But if we have e.g. 2000 voxels this determination is too time–consuming. Moreover, obviously in practice there cannot be a unique $\boldsymbol{x}$ which solves the LPP simultaneously for all $v \in V \setminus T$. So we will be content with another bound which is still good enough.

For each $v \in V \setminus T$ let $N(v)$ be a set of "neighboring" target voxels which "imply" a great dose value for v. E.g., $N(v)$ could contain from each CT–slice such a target voxel which has minimum Euclidean distance to v. Now it would be possible to replace the LPP by the new LPP

$$\begin{aligned} \boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{x} &\to \min \\ \boldsymbol{d}_w^{\mathrm{T}}\boldsymbol{x} &\geq b_T \text{ for all } w \in N(v) \\ \boldsymbol{x} &\geq \boldsymbol{0}. \end{aligned}$$

Here, the value of the objective function is in general smaller than for the old LPP since we have much more restrictions in the old LPP. But in order to save time we still want to avoid linear programming. Thus, to the new LPP, we add the restriction that all solutions $\boldsymbol{x}$ have only one non–zero component which then increases the minimal value $\underline{b}_v$ of the objective function (i.e., we assume that we have for v only one field in the treatment plan). It is easy to see that we have

$$\underline{b}_v = \min_{F \in \mathcal{F}} \max_{w \in N(v)} \frac{D_F(v)}{D_F(w)} b_T.$$

We call this bound the *minimum bound*. Considering for v not the best field but randomly (using a uniform distribution) any field of $\mathcal{F}$, we come to the *average bound*

$$\overline{b}_v = \frac{1}{|\mathcal{F}|} \sum_{F \in \mathcal{F}} \max_{w \in N(v)} \frac{D_F(v)}{D_F(w)} b_T.$$

Since in general $\underline{b}_v < \overline{b}_v$, the bound $\underline{b}_v$ has more influence to the WOP with respect to v than $\overline{b}_v$. In order to control this influence we use the general importance factors to finally fix the bound b_v. Let v belong to a region, say R, with general importance factor c_R (see Section 4). Then we put

$$b_v = \frac{1}{c_R + 1}\overline{b}_v + \frac{c_R}{c_R + 1}\underline{b}_v.$$

This value is a good estimate for the minimum possible dose at v. Hence we very often have (in particular if the general importance factor c_R is large) that

$$(\boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{x} - b_v)_+^2 = (\boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{x} - b_v)^2.$$

The piecewise quadratic objective function (1) is consequently more similar to a quadratic function than it would be in the case of general upper bounds b_R fixed by the physician. **An (almost) quadratic function can be handled much faster in the optimization process than a piecewise quadratic function**.

In addition, it is possible to finally multiply b_v by a factor not greater than 1 and decreasing with c_R. Then the corresponding voxel can still be better protected in the case of large importance factors, but this protection is at the cost of not reaching the prescribed dose of some target voxels.

6 Fast solution of the WOP

Essentially, all previously used methods for the solution of the WOP (see the references in the introduction) are variants of the scaled projection algorithm, [4]. But this algorithm has still much freedom, in particular in the choice of the scaling matrix. A standard classification is given by steepest descent, Jacobi, Gauss–Seidel, conjugate gradient, quasi–Newton, Newton. Sometimes it is difficult to extract from the literature which kind of algorithm is really used, how the projection is carried out and how the line–search is realized. We believe that it is indispensable to present precisely the concrete form of our algorithm. It is based on a projected Newton method of Bertsekas 1982 [3] adapted to the special kind of the objective function. So we work with the Hessian matrix, but in a relatively small dimension, similar to the active set approach (using cg) in [29]. Generally, authors hesitated to use the Hessian matrix, but it turned out that it does not pose numerical problems.

The objective function (1) reads briefly

$$f(\boldsymbol{x}) = \sum_{v \in T} c_v (\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_v)^2 + \sum_{v \in V \setminus T} c_v (\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_v)_+^2$$

where for the sake of simplicity

$$b_v = b_T \text{ for all } v \in T.$$

With

$$I_{\boldsymbol{x}} = T \cup \{v \in V \setminus T : \boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} > b_v\}$$

we have

$$f(\boldsymbol{x}) = \sum_{v \in I_{\boldsymbol{x}}} c_v (\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_v)^2. \tag{2}$$

Note that the gradient of f is given by

$$\nabla f(\boldsymbol{x}) = 2 \sum_{v \in I_{\boldsymbol{x}}} c_v (\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_v) \boldsymbol{d}_v = D\boldsymbol{x} - \boldsymbol{d} \tag{3}$$

where

$$D = 2 \sum_{v \in I_{\boldsymbol{x}}} c_v \boldsymbol{d}_v \boldsymbol{d}_v^{\mathrm{T}}$$

$$\boldsymbol{d} = 2 \sum_{v \in I_{\boldsymbol{x}}} c_v b_v \boldsymbol{d}_v.$$

In the algorithm we start with any nonnegative vector $\boldsymbol{x}$ (or with some heuristically good nonnegative $\boldsymbol{x}$ or with a solution of a previous WOP). We describe one step of the algorithm $\boldsymbol{x}_{old} \to \boldsymbol{x}_{new}$. The idea is to use a special *method of descent*. From the Karush–Kuhn–Tucker theorem, cf. [7], it follows that the admissible vector $\boldsymbol{x}$ is an optimal solution of the (convex) problem

$$f(\boldsymbol{x}) \to \min \text{ subject to } \boldsymbol{x} \geq \boldsymbol{0}$$

iff the KKT–conditions are satisfied:

$$\frac{\partial f}{\partial x_F} = 0 \text{ for all } F \in \mathcal{F} \text{ with } x_F > 0 \tag{4}$$

$$\frac{\partial f}{\partial x_F} \geq 0 \text{ for all } F \in \mathcal{F} \text{ with } x_F = 0. \tag{5}$$

(It is not difficult to verify this also directly without the KKT–theorem.) Thus, if the KKT–conditions are satisfied for $\boldsymbol{x} = \boldsymbol{x}_{old}$, the algorithm stops.

Now we assume that the KKT–conditions are not satisfied for the admissible vector $\boldsymbol{x} = \boldsymbol{x}_{old}$. In order to find an admissible direction of descent for $\boldsymbol{x}_{old}$ we first change a little bit the representation (2) of $f(\boldsymbol{x})$. Let $\boldsymbol{x}$ be any fixed vector, e.g. $\boldsymbol{x} = \boldsymbol{x}_{old}$. Let $\mathcal{F}' = \mathcal{F}'_{\boldsymbol{x}}$ be defined by

$$\mathcal{F}' = \left\{ F \in \mathcal{F} : x_F > 0 \text{ or } \frac{\partial f}{\partial x_F} < 0 \right\}. \tag{6}$$

The set $\mathcal{F}'$ contains all fields F for which the KKT–conditions (for $\boldsymbol{x}$) are not satisfied and moreover those fields F for which $x_F > 0$ and $\frac{\partial f}{\partial x_F} = 0$. We call $\mathcal{F}'$ the set of *free fields*. By our assumption, $\mathcal{F}' \neq \emptyset$. Note that

$$x_F = 0 \text{ for all } F \in \mathcal{F} \setminus \mathcal{F}'.$$

We are looking for a direction $\boldsymbol{z}$ which leaves the F–component equal to zero for all $F \in \mathcal{F} \setminus \mathcal{F}'$, i.e.

$$x_F + z_F = 0 \text{ for all } F \in \mathcal{F} \setminus \mathcal{F}'.$$

Such a direction could be the projected negative gradient $\boldsymbol{z}'$ (depending on $\boldsymbol{x}$) which is given by

$$z'_F = \begin{cases} -\frac{\partial f}{\partial x_F} & \text{if } F \in \mathcal{F}' \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

If we go along this direction, the index set $I_{\boldsymbol{x}}$ changes. For small $\lambda > 0$ we have for $v \in V \setminus T$

$$\boldsymbol{d}_v^{\mathrm{T}}(\boldsymbol{x} + \lambda \boldsymbol{z}') > b_v \text{ if } v \in I_{\boldsymbol{x}} \text{ or } \boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{x} = b_v \text{ and } \boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{z}' > 0.$$

Thus we define

$$I'_{\boldsymbol{x}} = I_{\boldsymbol{x}} \cup \{ v \in V \setminus T : \boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{x} = b_v \text{ and } \boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{z}' > 0 \} \tag{8}$$

and work with the new representation of $f(\boldsymbol{x})$

$$f(\boldsymbol{x}) = \sum_{v \in I'_{\boldsymbol{x}}} c_v (\boldsymbol{d}_v^{\mathrm{T}}\boldsymbol{x} - b_v)^2$$

(recall that $f(\boldsymbol{x})$ is not purely quadratic, because $I'_{\boldsymbol{x}}$ depends on $\boldsymbol{x}$).

With this new representation we start the determination of the admissible direction of descent for $\boldsymbol{x}_{old}$. For a moment, we consider two kinds of objects to be constant. Firstly, we have already said that we want to leave F–components of $\boldsymbol{x}$ equal to zero for $F \in \mathcal{F} \setminus \mathcal{F}'$. Thus let $\boldsymbol{x'}$ and $\boldsymbol{d'}_v$ be those subvectors of $\boldsymbol{x}$ and $\boldsymbol{d}_v$, respectively, whose components are associated with fields $F \in \mathcal{F}'$. This gives the function

$$g(\boldsymbol{x'}) = \sum_{v \in I'_{\boldsymbol{x}}} c_v (\boldsymbol{d'}_v^{\mathbf{T}} \boldsymbol{x'} - b_v)^2.$$

Secondly, we consider $I'_{\boldsymbol{x}}$ to be constant, i.e. $I'_{\boldsymbol{x}} = I'_{\boldsymbol{x}_{old}}$. This provides the new (purely quadratic) function

$$h(\boldsymbol{x'}) = \sum_{v \in I'_{\boldsymbol{x}_{old}}} c_v (\boldsymbol{d'}_v^{\mathbf{T}} \boldsymbol{x'} - b_v)^2$$

with the gradient

$$\nabla h(\boldsymbol{x'}) = D' \boldsymbol{x'} - \boldsymbol{d'}$$

where

$$D' = 2 \sum_{v \in I'_{\boldsymbol{x}_{old}}} c_v \boldsymbol{d'}_v \boldsymbol{d'}_v^{\mathbf{T}} \tag{9}$$

$$\boldsymbol{d'} = 2 \sum_{v \in I'_{\boldsymbol{x}_{old}}} c_v b_v \boldsymbol{d'}_v. \tag{10}$$

In practice, the vectors $\boldsymbol{x'}$ and $\boldsymbol{d'}_v$ often have few components. Thus the dimension of D' and $\boldsymbol{d'}$ is rather small which explains the high speed of the algorithm.

Let $\hat{\boldsymbol{x}}'$ be the optimal solution of

$$h(\boldsymbol{x'}) \to \min,$$

i.e., a solution of

$$D' \boldsymbol{x'} = \boldsymbol{d'}. \tag{11}$$

Let

$$\hat{\boldsymbol{z}}' = \hat{\boldsymbol{x}}' - \boldsymbol{x'}_{old}. \tag{12}$$

As above, the boldface dash in $\hat{\boldsymbol{z}}'$ means that its components are associated with fields $F \in \mathcal{F}'$, i.e. $\hat{\boldsymbol{z}}' = (\hat{z}_F)_{F\in\mathcal{F}'}$. Note that

$$D'\hat{\boldsymbol{z}}' = -\nabla h(\boldsymbol{x}'_{old}) \tag{13}$$

and that D' is the Hessian–matrix of h, i.e. $\hat{\boldsymbol{z}}'$ is a Newton–direction. As a direction of descent we finally take the vector $\boldsymbol{z}$ which is defined by

$$z_F = \begin{cases} \hat{z}_F & \text{if } (\hat{z}_F > 0 \text{ or } x_{old_F} > 0) \text{ and } F \in \mathcal{F}' \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Obviously, for small λ the vector $\boldsymbol{x}_{old} + \lambda\boldsymbol{z}$ remains nonnegative, hence $\boldsymbol{z}$ is an *admissible direction*. Before we continue with the description of the algorithm, we will show that $\boldsymbol{z}$ is indeed a direction of descent. Let

$$\varphi(\lambda) = f(\boldsymbol{x}_{old} + \lambda\boldsymbol{z}). \tag{15}$$

Note that φ is a piecewise quadratic convex function since f has, as a nonnegative combination of piecewise quadratic convex functions, the same property and that

$$\varphi'(\lambda) = \boldsymbol{z}^{\mathrm{T}}\nabla f(\boldsymbol{x}_{old} + \lambda\boldsymbol{z}).$$

Theorem 1 *If the KKT–conditions are not satisfied, then* $\varphi'(0) < 0$.

Proof We have to show that

$$\sum_{F\in\mathcal{F}'} z_F \frac{\partial f(\boldsymbol{x}_{old})}{\partial x_F} < 0. \tag{16}$$

Note that in view of $x_{old_F} = 0$ for all $F \in \mathcal{F} \setminus \mathcal{F}'$

$$\frac{\partial f(\boldsymbol{x}_{old})}{\partial x_F} = \frac{\partial g(\boldsymbol{x}'_{old})}{\partial x_F} = \frac{\partial h(\boldsymbol{x}'_{old})}{\partial x_F} \text{ for all } F \in \mathcal{F} \setminus \mathcal{F}'.$$

Since by supposition the KKT–conditions are not satisfied

$$\nabla h(\boldsymbol{x}'_{old}) \neq \boldsymbol{0}$$

which establishes $\boldsymbol{x}'_{old} \neq \hat{\boldsymbol{x}}'$ and moreover $h(\hat{\boldsymbol{x}}') < h(\boldsymbol{x}'_{old})$. Consequently, $\hat{\boldsymbol{z}}' \neq \boldsymbol{0}$ and $\hat{\boldsymbol{z}}'$ is a direction of descent for h, i.e.

$$\hat{\boldsymbol{z}}'^{\mathrm{T}}\nabla h(\boldsymbol{x}'_{old}) < 0$$

which is the same as

$$\sum_{F\in\mathcal{F}'} \hat{z}_F \frac{\partial f(\boldsymbol{x}_{old})}{\partial x_F} < 0. \tag{17}$$

We show that for all $F \in \mathcal{F}'$

$$\hat{z}_F \frac{\partial f(\boldsymbol{x}_{old})}{\partial x_F} \geq z_F \frac{\partial f(\boldsymbol{x}_{old})}{\partial x_F}. \tag{18}$$

This is clear if $z_F = \hat{z}_F$. If $z_F = 0 \neq \hat{z}_F$, in view of (14) necessarily $\hat{z}_F \leq 0$ and $x_{old_F} = 0$. By the definition of $\mathcal{F}'$

$$\frac{\partial f(\boldsymbol{x}_{old})}{\partial x_F} < 0$$

and hence (18) is satisfied. From (17) and (18) we obtain (16). ■

Now, having found the direction of descent $\boldsymbol{z}$, we have to describe how far we are going along this direction. This part of the algorithm is called *one–dimensional minimization*. We will not go farther than to $\hat{\boldsymbol{x}}$ and, moreover, we have to leave all components nonnegative. Thus let

$$\lambda_{\max} = \min\{1, \max\{\lambda : \boldsymbol{x}_{old} + \lambda \boldsymbol{z} \geq \boldsymbol{0}\}\}. \tag{19}$$

If $\varphi'(\lambda_{\max}) \leq 0$, φ is still non–increasing at $\lambda = \lambda_{\max}$ and thus we put

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{old} + \lambda_{\max} \boldsymbol{z}.$$

Here the one-dimensional minimization is terminated and one iteration step of the algorithm is completed.

If $\varphi'(\lambda_{\max}) > 0$, the minimum of φ lies in the open interval $(0, \lambda_{\max})$. If φ was purely quadratic (i.e. if f had no items of the form $(\boldsymbol{d}_v^{\mathrm{T}} \boldsymbol{x} - b_v)_+^2$), then φ and φ' would be a quadratic and linear real function, respectively. It is easy to calculate the root λ_0 of $\varphi'(\lambda)$ for this case, i.e. the minimum λ_0 of $\varphi(\lambda)$:

$$\lambda_0 = -\frac{\lambda_{\max} \varphi'(0)}{\varphi'(\lambda_{\max}) - \varphi'(0)}.$$

We would then put

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{old} + \lambda_0 \boldsymbol{z}.$$

But f and thus also φ are only very similar to a quadratic function (by our special choice of the upper bounds b_v). In reality f and φ are piecewise quadratic, convex, differentiable functions. Consequently, λ_0 could lie in another piece of φ than 0 or $\lambda_{\max}$ and possibly $\varphi'(\lambda_0) \neq 0$. Hence we substitute

$$\lambda_{\max} := \lambda_0$$

and continue with the case distinction $\varphi'(\lambda_{\max}) \gtreqless 0$ as before and iterate. Since we have only finitely many pieces of φ, after a finite number of such substitutions the case $\varphi'(\lambda_{\max}) \leq 0$ must occur which shows that in finite time one iteration step of the algorithm is completed. In numerical tests the iteration was not necessary in most cases since the situation $\varphi'(\lambda_{\max}) \leq 0$ was already in the beginning. In particular, it is not necessary to use other line–search methods like golden section, Fibonacci, or descent rules using derivatives [7, 22]. We summarize the whole procedure:

Algorithm WOP

Fix a starting vector $\boldsymbol{x} \geq \boldsymbol{0}$
Determine $\nabla f(\boldsymbol{x})$ by (3)
While the KKT–conditions (4) and (5) are not satisfied do
    Determine $\mathcal{F}'$ by (6)
    Determine $\boldsymbol{z}'$ by (7)
    Determine $I'_{\boldsymbol{x}}$ by (8)
    Determine D' and $\boldsymbol{d'}$ by (9) and (10)
    Determine $\boldsymbol{z}$ by (13) and (14)
    Put $\varphi(\lambda) = f(\boldsymbol{x} + \lambda \boldsymbol{z})$ as in (15)
    Determine $\lambda_{\max}$ by (19)
    Determine $\nabla f(\boldsymbol{x} + \lambda_{\max} \boldsymbol{z})$ by (3)
    while $\varphi'(\lambda_{\max}) > 0$ do
        $\lambda_{\max} = -\lambda_{\max} \varphi'(0) / (\varphi'(\lambda_{\max}) - \varphi'(0))$
        Determine $\nabla f(\boldsymbol{x} + \lambda_{\max} \boldsymbol{z})$ by (3)
    Put $\boldsymbol{x} = \boldsymbol{x} + \lambda_{\max} \boldsymbol{z}$

Clearly, some numerical precautions as the replacement of 0 by a small ε must be included.

We tested also the use of other directions of descent, e.g. the projected negative gradient and a variant of a projected conjugate gradient (two–dimensional minimization in the subspace spanned by the projected gradient and the projection of the last direction). However, the variant presented

above in detail was the best one. The most time–consuming parts are the determination of the gradient of f in (3) and the determination of the matrix D' in (9) (not the solution of the system (13)!). A slight acceleration can be obtained if one replaces the Hessian–matrix D' by a good estimation of D' (non–standard quasi–Newton method). This can be done as follows: Choose with probability p voxels v from $I'_{\boldsymbol{x}_{old}}$ which gives the set $I^p_{\boldsymbol{x}_{old}}$. Then replace (9) by

$$D' = \frac{2}{p} \sum_{v \in I^p_{\boldsymbol{x}_{old}}} c_v \boldsymbol{d'}_v \boldsymbol{d'}^{\mathbf{T}}_v.$$

But numerical tests show that p cannot be taken small because otherwise the number of iterations increases so much that the whole algorithm is not faster. A good choice is e.g. $p = 0.5$.

7 One–field–optimization

For time and memory reasons it is useful to fix several field parameters independently of the other fields with simple search strategies. By heuristic reasons we do the following:

- Place the isocenter at the center of gravity of the PTV.
- Determine field width and field length in such a way that the jaws touch a "security strip" around the PTV and do the same for the leaves of an MLC. At this stage, in the beam's eye view, it remains an open area which contains the whole PTV and is sufficiently small.
- Determine the field angle such that the open area has minimum size.
- Determine for each energy the quotient of a good estimate of the ratio of the total dose accumulated at the PTV and the total dose accumulated at all OARs (e.g. compute the dose of several randomly chosen voxels). Take that energy which yields the greatest ratio.

We emphasize that it is possible to determine the energy also globally like the gantry angle, table angle and kind of wedge. There is no theoretical and practical obstacle. The only problem is that this increases time and memory demand. Tests have shown that global determination of energy does not

have such great influence as global determination of angles and wedges. Thus we described the one–field–energy–optimization here. The algorithms for all items of this section are straightforward, hence we omit a detailed discussion.

8 Field management

Because of our one–field–optimization procedure we have finally only 4 essential field parameters: gantry angle $F \to \varphi$, table angle $F \to \theta$, kind of wedge $F \to \kappa$ and the distinguished parameter x_F. We assume that there is a set Φ of g possible gantry angles $\Phi = \{\varphi_0, \ldots, \varphi_{g-1}\}$ (usually $\Phi = \{0, \ldots, 359\}$), a set Θ of t possible table angles $\Theta = \{\theta_0, \ldots, \theta_{t-1}\}$ (usually $\Theta = \{0, \ldots, 179\}$), and a set $\mathcal{K}$ of w possible kinds of wedges $\mathcal{K} = \{\kappa_0, \ldots, \kappa_{w-1}\}$ (e.g. $\mathcal{K} = \{0, \ldots, 4\}$, where 0 means no wedge and 1–4 means a wedge in one of 4 positions differing in rotations of 90 degrees). From all these triples $(\varphi, \theta, \kappa)$ often not all are allowed. Firstly, there are technical reasons and secondly there are reasons to forbid several triples. For example, one should forbid those angle pairs (φ, θ) which yield a small angle between the central axis of the patient and the central ray, because in these cases the ray runs through the whole body and there are not enough CT–slices for the dose calculation. Moreover, table–gantry–collisions may occur. Thus let

$$\mathcal{A} \subseteq \Phi \times \Theta \times \mathcal{K}$$

be the set of *allowed triples*. We call a field F *allowed* if $(F \to \varphi, F \to \theta, F \to \kappa) \in \mathcal{A}$. Now we could apply our Algorithm WOP to

$$\mathcal{F} = \{F : (F \to \varphi, F \to \theta, F \to \kappa) \in \mathcal{A}\}$$

But in our example $\Phi \times \Theta \times \mathcal{K}$ has $360 \cdot 180 \cdot 5 = 324,000$ elements. Thus the set $\mathcal{F}$ of allowed fields is also very large. For each such triple the one–field–optimization should be carried out. Having moreover, e.g. 2000, voxels we would need 648,000,000 dose calculations. This is too time–consuming and poses memory problems. Thus we first restrict ourselves to a subset $\mathcal{A}'$ of the set $\mathcal{A}$ of all allowed triples whose elements are uniformly distributed in $\mathcal{A}$: Let $\Phi' \subseteq \Phi, \Theta' \subseteq \Theta$, $\mathcal{K}' \subseteq \mathcal{K}$, and let $\mathcal{A}' := (\Phi' \times \Theta' \times \mathcal{K}') \cap \mathcal{A}$. For example, if $\Phi' = \{0, 30, 60, \ldots, 330\}, \Theta' = \{0, 30, \ldots, 150\}, \mathcal{K}' = \{0, 1, 2, 3, 4\}$, then $|\Phi' \times \Theta' \times \mathcal{K}'| = 360$ and therefore $|\mathcal{A}'| \leq 360$. Here the angles are uniformly

distributed and "roughly all" directions are possible. We first solve the WOP for the set

$$\mathcal{F}' = \{F : (F \to \varphi, F \to \theta, F \to \kappa) \in \mathcal{A}'\}. \tag{20}$$

Now the main idea is the following: **We do not start with $\mathcal{F}'$. Instead, we work with an increasing sequence of subsets of the set $\mathcal{A}'$ whose elements are again uniformly distributed in $\mathcal{A}$. The optimal solution for an element of the sequence (i.e. for a subset of allowed triples) can be taken as the starting vector for the next element of the sequence. Then at each moment the number of fields having nonzero weight remains very small and this yields high speed in the solution of the WOP.**

Formally, we describe this as follows. Keeping in mind uniform distributions of the angles, we fix sequences $\Phi_0 \subseteq \Phi_1 \subseteq \cdots \subseteq \Phi_s = \Phi'$, $\Theta_0 \subseteq \Theta_1 \subseteq \cdots \subseteq \Theta_s = \Theta'$, $\mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \cdots \subseteq \mathcal{K}_s = \mathcal{K}'$, we put $\mathcal{A}_i := (\Phi_i \times \Theta_i \times \mathcal{K}_i) \cap \mathcal{A}$ and solve the WOP for the sets

$$\mathcal{F}_i = \{F : (F \to \varphi, F \to \theta, F \to \kappa) \in \mathcal{A}_i\}, i = 0, \ldots, s.$$

Note that $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_s$. The aforementioned advantage is that we can take the optimal solution $\boldsymbol{x}_i$ for $\mathcal{F}_i$ as a starting vector for $\mathcal{F}_{i+1}$ where we put

$$x_F = 0 \text{ for all } F \in \mathcal{F}_{i+1} \setminus \mathcal{F}_i, i = 0, \ldots, s-1.$$

This yields the small number of nonzero components which essentially influences the speed. Many experiments have shown that still in the optimal WOP–solution $\boldsymbol{x}'$ for $\mathcal{F}'$ there are only a few nonzero components which can be considered to represent roughly the significant fields. So we first delete all non–significant fields by putting

$$\mathcal{F}'' := \{F \in \mathcal{F}' : x_F > 0\}.$$

Reducing correspondingly $\boldsymbol{x}'$ yields $\boldsymbol{x}''$.

Hitherto we did not allow all triples from $\mathcal{A}$, but only the triples from $\mathcal{A}'$. Now we essentially allow all triples from $\mathcal{A}$ using an analogue procedure (we avoid introducing further subsets of $\mathcal{A}$, but work immediately with sets of fields and consider here only angles): **In the neighborhood of the remaining significant fields, we permit step by step more fields until all fields corresponding to triples from $\mathcal{A}$ are allowed.** Again

more formally: Let h_1 and h_2 be the difference between neighboring angles in Φ' and Θ', in our example we have $h_1 = 30$ and $h_2 = 30$. Now we add to $\mathcal{F}''$ those (allowed) fields G for which there exists a field F in $\mathcal{F}''$ differing from G only in the gantry angle by $h_1/2 \pmod{360}$, i.e. we put

$$\begin{aligned}\mathcal{F}''' = \mathcal{F}'' \cup \{G : G \text{ is allowed and there is some } F \in \mathcal{F}'' \text{ such that} \\ F \to \varphi - G \to \varphi = \pm h_1/2 \pmod{360} \text{ and} \\ F \to \theta = G \to \theta, F \to \kappa = G \to \kappa\}.\end{aligned}$$

Starting with $\boldsymbol{x}''$, the Algorithm WOP provides a solution $\boldsymbol{x}'''$. Then we proceed in the same way for table angles:

$$\begin{aligned}\mathcal{F}^{(iv)} = \mathcal{F}''' \cup \{G : G \text{ is allowed and there is some } F \in \mathcal{F}''' \text{ such that} \\ F \to \varphi \doteq G \to \varphi, F \to \theta - G \to \theta = \pm h_2/2, F \to \kappa = G \to \kappa\}.\end{aligned}$$

Here the θ–difference is considered $\pmod{180}$. If in this construction $G \to \theta$ becomes less than 0 or greater or equal to 180 then we have to add or subtract 180 from this angle. Reversing the direction of the table requires also the reflection of the gantry angle with respect to a vertical line in order to get the same beam–direction for the patient. Hence in that case we must replace the gantry angle $G \to \varphi$ by $360 - G \to \varphi$. Because of this additional condition we write $F \to \varphi \doteq G \to \varphi$ instead of $F \to \varphi = G \to \varphi$.

We start the Algorithm WOP with $\boldsymbol{x}'''$ and finally obtain $\boldsymbol{x}^{(iv)}$. Now $h_1/2$ and $h_2/2$ still may be too large. Thus we put

$$\mathcal{F}' := \mathcal{F}^{(iv)}, \boldsymbol{x}' := \boldsymbol{x}^{(iv)}, h_1 := h_1/2, h_2 := h_2/2$$

and iterate until we obtain the desired refinement. Again, deleting the nonzero components, at the end we have a set $\mathcal{F}^*$ with an associated vector $\boldsymbol{x}^*$ that has no zero component. The pair $(\mathcal{F}^*, \boldsymbol{x}^*)$ can be considered as the *optimal treatment plan* though we used little heuristics (restricting at the end only to neighbors of significant fields) in order to avoid to work with all allowed triples from $\Phi \times \Theta \times \mathcal{K}$.

9 Field reduction

The treatment plan $(\mathcal{F}^*, \boldsymbol{x}^*)$ obtained so far has for concrete patients often between 15 and 30 fields. This is for practical purposes too much. Assume

that the objective function has for $(\mathcal{F}^*, \boldsymbol{x}^*)$ a value z^*. We are looking for a treatment plan $(\mathcal{F}, \boldsymbol{x})$, having z as the value of the objective function, for which

$$\frac{z}{z^*} \leq 1 + \varepsilon$$

where ε is a small positive number, e.g. 0.1. Such a treatment plan is still good enough, i.e. ε–approximatively optimal. The idea is trivial: **Delete step by step one or simultaneously several fields from $\mathcal{F}^*$ and update the optimal solution $\boldsymbol{x}$ using the Algorithm WOP**. For the deletion we choose such (non–significant) fields F for which $x_F \sum_{v \in T} D_F(v)$, i.e. the total dose contribution to the PTV, is small. If $\mathcal{F}^*$ is large, the optimal value of the objective function still remains almost constant in the beginning of this process. But if the actual treatment plan has a few fields after a while, this fast deletion heuristics is not good enough. So at some moment we start new, slower reduction heuristics consisting of two steps:

1. **Greedy deletion:** Running through all fields we determine that field F which yields the smallest optimal value of the objective function after its deletion. This field will be deleted.

2. **Local search:** After deletion of the most unimportant field, the remaining set $\mathcal{F}_{old}$ of fields is generally not the best one compared with all sets of fields of same cardinality. But one can expect that only a small adjustment is necessary to get again a new, really good set $\mathcal{F}_{new}$ of fields (adjustment only at the end, i.e. after greedily deleting many fields, would pose much more problems with local optima and e.g. time–consuming simulated annealing would be necessary). The adjustment will be done by a special kind of local search. Suppose that $\mathcal{F}_{old} = \{F_1, \ldots, F_n\}$. With each field F_i we associate the set of neighboring fields

$$\begin{aligned} N(F_i) = \{G : G \text{ is allowed }, F_i \to \kappa = G \to \kappa, \text{ and} \\ (F_i \to \varphi - G \to \varphi = \pm h \text{ and } F_i \to \theta = G \to \theta) \text{ or} \\ (F_i \to \varphi \doteq G \to \varphi \text{ and } F_i \to \theta - G \to \theta = \pm h) \end{aligned}$$

where h is some distance, e.g. $h = 1$ or 2, and the subtraction is $\pmod{360}$ or $\pmod{180}$ analogously as at the end of Section 8. In general, $|N(F_i)| = 5$, $i = 1, \ldots, n$. We put

$$\tilde{\mathcal{F}} = \bigcup_{i=1}^{n} N(F_i)$$

and, starting with the actual solution, we apply the Algorithm WOP to $\tilde{\mathcal{F}}$ which gives the solution $\tilde{\boldsymbol{x}}$. Deleting from $\tilde{\mathcal{F}}$ all fields with $\tilde{x}_F = 0$ could provide a set of fields which has more elements than $\mathcal{F}_{old}$. Hence we would not have a field reduction and we would be unsuccessful. Consequently, we do the following: For each $i, i = 1, \ldots, n$, we look for the most significant field $\tilde{F}_i$ in $N(F_i)$, i.e. for which $x_F \sum_{v \in T} D_F(v)$ is maximal, $F \in N(F_i)$. Then we put

$$\mathcal{F}_{new} = \{\tilde{F}_1, \ldots, \tilde{F}_n\}.$$

If for $i \neq j$, $N(F_i) \cap N(F_j) \neq \emptyset$, it is possible that $\tilde{F}_i = \tilde{F}_j$ and hence $|\mathcal{F}_{new}| < |\mathcal{F}_{old}|$, but on the one hand this only accelerates the reduction process and on the other hand in practical tests this never appeared. Let z_{old} and z_{new} be the optimal values of the objective function for $\mathcal{F}_{old}$ and $\mathcal{F}_{new}$, respectively. If $z_{old} > z_{new}$, we replace $\mathcal{F}_{old}$ by $\mathcal{F}_{new}$ and iterate. At some moment we will have the situation $z_{old} \leq z_{new}$; then we do not replace and the reduction step is completed.

The field reduction is now carried out as long as the desired number of fields is reached. The ratio z/z^* gives a hint, how much the real world patient has to "pay" for being treated with a smaller than the optimal number of fields. But we emphasize that e.g. a ratio of 2 does by no means say that the probability of being cured is divided by 2.

10 Choice of voxels

The most time–consuming assignments (3) and (9) show that the algorithm WOP, and thus also the whole algorithm is more or less time–proportional to the number of voxels. Hence we have to make a good choice of voxels. Random choice of voxels was used e.g. in [32, 40]. We propose to consider 4 *types* of voxels and to associate with them probabilities $p_0 > p_1 > p_2 > p_3$, e.g. $p_0 = 1, p_1 = 1/2, p_2 = 1/4, p_3 = 1/16$. A voxel v is of type 0 or 1 if it is on the geometrical boundary of the PTV T or of an OAR $R \in \mathcal{R}$, respectively. The voxels on the boundary seem to have most influence to the treatment plan. A voxel is of type 2 if it is in the interior of the PTV or of an OAR. Moreover, voxels of the rest S which are "near" to the PTV are also considered as voxels of type 2. Finally, all remaining voxels of S are of type 3 .

In addition, we distinguish visible and non–visible voxels. A voxel v is *visible* iff there is at least one field F in the starting set $\mathcal{F}'$ (20) such that v is not covered in the beam's eye view by the jaws or the leaves given by F.

Running through the complete set V of voxels, we select visible voxels of type i with probability $p_i, i = 0, 1, 2, 3$. This gives the working set V' of voxels for which the optimization process is really carried out.

11 PTV–oriented placement of the leaves of an MLC

The *MLC* enables each field F to open a certain region which can be described as follows: There is given a rectangle of size $(F \to W) \times (F \to L)$. This rectangle is divided into m inner–point–disjoint subrectangles having size $(F \to W) \times h_i$, $i = 1, \ldots, m$, where $\sum_{i=1}^{m} h_i = F \to L$. For each subrectangle there are given values l_i and r_i, $i = 1, \ldots, m$, which indicate the positions of the edges of the left and right leaf. Let $\boldsymbol{l} = (l_1, \ldots, l_m)$ and $\boldsymbol{r} = (r_1, \ldots, r_m)$. The *shape* of the field is consequently given as a quadruple $(F \to W, F \to L, \boldsymbol{l}, \boldsymbol{r})$ and collects the field parameters (2)–(4). This is illustrated in the left part of Figure 2.

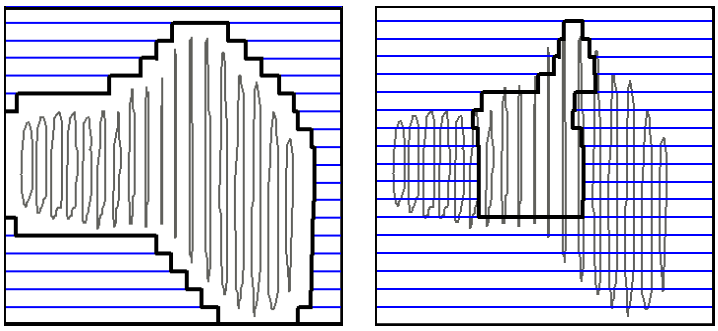

Figure 2: Field–shape (left) and field shape with $P = ((1, 2), (1, 3))$ (right)

Up to now we determined $\boldsymbol{l}$ and $\boldsymbol{r}$ in such a way that the leaves touch a "security strip" around the PTV (see Section 7), but note that for real world MLCs the parameters $F \to W$ and $F \to L$ are fixed and in particular greater

than the corresponding results in the one–field–optimization. From now on we allow that parts of the PTV are covered by leaves. We already mentioned that this is particularly interesting for non–convex PTV's.

The standard way is the discretization of the leaf–positions. This leads to a partition of the rectangle into several subrectangles (resp. subsquares) which are irradiated by "pencil–beams". In conventional algorithms, the intensity (relative fluence) is calculated for every subrectangle during the optimization process. Then these intensities can be realized as a superposition of several leaf–positions which are calculated by a leaf–segmentation algorithm, cf. [44]. But since one has a large number of subrectangles the computation must be restricted to a small number of allowed beam–orientations. On the contrary, our aim is the simultaneous optimization of the beam–orientations, i.e. of the angles, and of the intensities. So we work out two ideas: **The discretization should be done relative to the PTV, it should not be too fine and larger open regions should be allowed in order to avoid a subsequent leaf–segmentation.**

For this purpose, we will introduce a notational system so that we will be able to represent certain subregions of the PTV efficiently. To motivate this notation, consider the field–shapes shown in Figure 2. The shape on the right is a subregion of the field on the left formed in a very specific way as follows. First, we divide the nonempty rows up into $p = 3$ groups of roughly equal size (group 0 = rows 0–5, group 1 = rows 6–10, and group 2 = rows 11–16). We then completely close leaves in group 0. More generally, we would close off groups at the beginning, end, or both, leaving open one or more contiguous groups. To denote which groups are left open, we use the notation $(i_2, j_2) = (1, 3)$, which corresponds to groups i_2 to $j_2 - 1$ being left open. For the rows that are not closed off, the opening in each row is divided into $p = 3$ roughly equal subintervals, and the first and third subintervals are closed off, leaving only the middle subinterval open. More generally, subintervals can be closed off at the beginning, the end, or both, leaving one or more contiguous subintervals open. The notation $(i_1, j_1) = (1, 2)$ is used to indicate that subintervals in the range i_1 to $j_1 - 1$ are left open.

Now, in order to describe this procedure formally, we introduce the following *basic positions*: Let λ and ρ be defined by

$$\lambda = \min\{i \in \{1, \ldots, m\} : l_i < r_i\}$$
$$\rho = \max\{i \in \{1, \ldots, m\} : l_i < r_i\}.$$

Thus λ (resp. ρ) is the first (resp. last) number of a pair of leaves which do not cover the corresponding subrectangle completely. We consider $\boldsymbol{l}$ and $\boldsymbol{r}$ as fixed (given by one–field–optimization) and describe new positions with two pairs $(i_1, j_1), (i_2, j_2)$ where $0 \le i_1 < j_1 \le p, 0 \le i_2 < j_2 \le p$ and p is some fixed number, say $p = 3$ or 4. Let briefly

$$P = ((i_1, j_1), (i_2, j_2)).$$

Under the leaf–pairs with indices $\lambda, \ldots, \rho$ we close a certain number completely, namely: We divide the interval $[\lambda, \rho]$ into p almost equal parts of average length $(\rho - \lambda)/p$ and we completely close exactly the parts $0, \ldots, i_2 - 1, j_2, \ldots, p-1$. The remaining leaf–pairs which do not completely cover the corresponding rectangle have indices $\underline{k}, \ldots, \overline{k}$, where $\underline{k}$ and $\overline{k}$ are integers near to $\lambda + i_2(\rho - \lambda)/p$ and $\lambda + j_2(\rho - \lambda)/p$, respectively, more precisely,

$$\underline{k} = \begin{cases} \left\lfloor \frac{(p-i_2)\lambda + i_2\rho}{p} \right\rfloor + 1 & \text{if } i_2 > 0 \\ \lambda & \text{if } i_2 = 0 \text{ and } j_2 > 0 \end{cases}$$

$$\overline{k} = \left\lfloor \frac{(p - j_2)\lambda + j_2\rho}{p} \right\rfloor$$

For each such remaining leaf–pair with index $k \in [\underline{k}, \overline{k}]$ we describe new positions of the leaves in the following way: The open region of the corresponding subrectangle is divided into p equal parts, each of length $(r_k - l_k)/p$. The part i_1 starts at position $l_k + i_1(r_k - l_k)/p$ and the part $j_1 - 1$ ends (and the part j_1 starts) at position $l_k + j_1(r_k - l_k)/p$. We move the leaves in such a way, that the parts $i_1, \ldots, j_1 - 1$ remain open. Formally this means that we associate with P the new vectors $\boldsymbol{l}^P$ and $\boldsymbol{r}^P$ as follows: Let for $k = 1, \ldots, m$

$$l_k^P = \begin{cases} \frac{(p-i_1)l_k + i_1 r_k}{p} & \text{if } \underline{k} \le k \le \overline{k} \\ l_k & \text{otherwise} \end{cases}$$

$$r_k^P = \begin{cases} \frac{(p-j_1)l_k + j_1 r_k}{p} & \text{if } \underline{k} \le k \le \overline{k} \\ l_k & \text{otherwise.} \end{cases}$$

If the shape $(F \to W, F \to L, \boldsymbol{l}, \boldsymbol{r})$ is considered as a distorted rectangle, then $(F \to W, F \to L, \boldsymbol{l}^P, \boldsymbol{r}^P)$ can be considered as a distorted subrectangle. In order to be able to work with such fields we assign with each field F the 4 new *shape–parameters*

(10) $F \to i_1$
(11) $F \to j_1$
(12) $F \to i_2$
(13) $F \to j_2$.

If the parameters (1)–(9) of a field are fixed, then variation of the parameters (10)–(13) yields "dependent" fields. One could think that it would be enough to restrict to parameters where $j_1 - i_1 = 1$ and $j_2 - i_2 = 1$. But firstly one field with a large open region is better than a superposition of many fields with inner–point–disjoint small open regions because also leaf–covered voxels get a certain small dose which can be in the sum large, and secondly numerical tests have shown that the admission of this larger set of dependent fields numerically saves time and behaves better.

In this approach we consider the leaves for each field as static. If movements of the leaves are allowed (dynamic case) then adequate combinations of our basic fields should be taken as new basic fields and the optimization process must be carried out for this new basic set.

If $p = 3$ or 4 we have already 36 or 100 choices for the shape–parameters (recall $0 \le i_1 < j_1 \le p, 0 \le i_2 < j_2 \le p$). Hence, if we try to optimize globally all field parameters (7)–(13) instead of only parameters (7)–(9), then the starting set $\mathcal{F}'$ in (20) (here given by 7–tuples instead of triples) is much larger than before. We have two ways out:

1. **Slow method:** We first admit only fields with $i_1 = 0, j_1 = p$ or $i_2 = 0, j_2 = p$ and later we admit also other pairs (i_2, j_2) using a refinement procedure like the angle–refinement from Section 8.

2. **Fast method:** We first optimize the parameters (1)–(9) as described in Sections 4–10 and obtain the treatment plan $(\mathcal{F}, \boldsymbol{x})$. For each field $F \in \mathcal{F}$ we have here

$$F \to i_1 = F \to i_2 = 0 \text{ and } F \to j_1 = F \to j_2 = p.$$

Then we extend $\mathcal{F} = \mathcal{F}_0$ step by step as in the beginning of Section 8. We put for $i = 1, \ldots, p-1$

$$\mathcal{F}_i = \{F : F \to j_1 - F \to i_1 \ge p - i \text{ and } F \text{ coincides in all other parameters with some field of } \mathcal{F}\}.$$

We obtain a sequence $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_{p-1}$. Further, we put for $i =$

$1, \ldots, p-1$

$$\mathcal{F}_{p-1+i} = \{F : F \to j_2 - F \to i_2 \geq p - i \text{ and } F \text{ coincides in all other parameters with some field of } \mathcal{F}_{p-1}\}.$$

Again, we obtain a sequence $\mathcal{F}_{p-1} \subset \mathcal{F}_p \subset \cdots \subset \mathcal{F}_{2p-2}$. An optimal vector $\boldsymbol{x}_i$ for $\mathcal{F}_i$ can be taken as a starting vector for Algorithm WOP applied to $\mathcal{F}_{i+1}, i = 0, \ldots, 2p-3$. With the field reduction from Section 9 we ultimately obtain the desired treatment plan.

One must find a good compromise between the number of beam–orientations and the number of fields (recall that we consider equal beam–orientations but different leaf–positions as different fields). The best results can be obtained if each field has its own beam–orientation. In this case there cannot arise tongue and groove effects that cause underdosage. But nowadays a change of the table angle needs more time than a change of the leaf–position, hence one should restrict to a small number of beam–orientations. This can be obtained e.g. by the aforementioned fast method. Also here underdosage is not significant because we are working in advance with large open fields and for each beam–orientation the number of fields is small.

Moreover we mention that for the basic positions in almost all cases the interleaf collision constraint (forbidding collision of neighboring leaves) is not violated because the open region in a basic position is, in some sense, similar to the PTV. But if this once occurs one only has to modify the concrete basic position slightly (shifting a leaf a little bit back) in order to avoid collision. This modification does not have significant influence to the algorithm.

Finally we recall (see Section 2) that fields which differ only in the leaf–setting and the weight can be considered as one intensity modulated field, only. With a leaf–setting algorithm, cf. [44], one can try to realize this field in a better way, i.e. with fewer fields (segments) or with less total time. New such algorithms are given in [20, 30].

12 PTV– and OAR–oriented placement of the leaves of an MLC

In the last section we determined basic fields independently of the OARs. Here we present another, more geometric method. Suppose, some beam

orientation is fixed and suppose that we have r OARs $R_1, \ldots, R_r$. Then we introduce $2r + 1$ basic fields as follows: F_0 is the field as in the beginning of Section 11, where the leaves touch a security strip around the PTV. F_i (respectively F_{r+i}), $i = 1, \ldots, r$, are those fields which can be obtained from F_0 by shifting the left leaves to the right (respectively the right leaves to the left) as few as possible such that the OAR R_i is just covered.

In the field management, we first only allow fields of type F_0, then fields of type F_0, F_1 and so on until finally all types are allowed. With this method e.g. the rectum and the spinal cord can be protected in a good way.

13 Results

The aim of the paper is the detailed mathematical presentation of the algorithm. A complete description of results for several kinds of patients would lie beyond the scope of the paper and thus we postpone this to further, more clinically oriented publications. We tested the algorithm already for many patients. Qualitatively, the following statements can be made. The results are significantly better if one uses

- optimized angles instead of equidistant angles (also in the case of an MLC),
- non–coplanar beams instead of coplanar beams,
- compensators or MLCs instead of standard rectangular fields.

The placement of the leaves of an MLC described in Sections 11 and 12 plays an essential role only in the case of non–convex PTVs. We emphasize, that the optimization of the beam–orientation is more important than the optimization of intensity maps of fields with equidistant angles.

The field reduction process from Section 9 can be carried out up to a relatively small number of fields without significantly increasing the objective function. A steep ascent of this function starts, depending on the patient, with about 4 to 8 fields.

In the optimal solution almost all fields contain wedges. This shows that one may better prevent a dose–decrease in the boundary region of the PTV by means of wedges.

14 Concluding remarks

Clearly, the time for the whole optimization process depends on the patient and on the choice of several algorithm parameters which can be easily adjusted. The time is in the range from 20 seconds to 5 minutes on a PC with a 1.8 GHz processor and 256 MB RAM. High speed can be obtained via an efficient implementation. In particular, the dose values $D_F(v)$ should be only computed if they are really needed and they should be stored as long as they are needed. Thus a good interaction between optimization and dose calculation is necessary.

It is not a problem to replace the dose calculation method by another method, if the method is not essentially slower. But e.g. a factor of 10 is still completely practicable. If the factor is greater than 100 the other method should be used only at the end of the optimization process.

In our version the most time–consuming steps are the determination of the coordinates of the dose calculation point in the beam's–eye–view coordinate system having the central ray as one axis and the determination of the (density weighted) distances which are covered by the ray resp. central ray.

Our module can be included into a complete treatment planning system without problems. Improvements in the conformity of the dose distribution and in the protection of normal tissue provides higher tumor control and fewer complications.

Acknowledgement. The authors are grateful to an anonymous referee whose comments helped to improve the presentation of the paper.

References

[1] Aspradakis M M and Redpath A T 1997 A technique for the fast calculation of three–dimensional photon dose distributions using the superposition model *Phys. Med. Biol.* **42** 1475–89

[2] Bahr G K, Kereikas J G, Horwitz H, Finney R, Galvin J and Goode K 1968 The method of linear programming applied to radiation treatment planning *Radiology* **91** 686–93

[3] Bertsekas D 1982 Projected Newton methods for optimization problems with simple constraints *SIAM J. Control Opt.* **20** 221–46

[4] Bertsekas D P and Tsitsiklis J N 1989 *Parallel and distributed computation* (Englewood Cliffs, NJ: Prentice–Hall, Inc.)

[5] Birkenhagen U, Bollmann R, Schmidt K-P, Tabbert E and Weigel G 1997 Verifikation der Dosisberechnung des Bestrahlungsplanungssystems ProPlan mittels Thermolumineszensdosimetrie und Alderson–Phantom *Z. Med. Phys.* **7** 124–9

[6] Bollmann R, Schmidt K-P and Tabbert E 1981 Verbesserung der Bestrahlungsplanung in der Hochvolttherapie duch mathematische Optimierung *Radiobiol. Radiother.* **5** 594–601

[7] Bomze I M and Grossmann W 1993 *Optimierung–Theorie und Algorithmen* (Mannheim: BI–Wissenschaftsverlag)

[8] Bortfeld T, Burkelbach J, Boesecke R and Schlegel W 1990 Methods of image reconstruction from projections applied to conformation radiotherapy *Phys. Med. Biol.* **35** 1423–34

[9] Bortfeld T and Schlegel W 1993 Optimization of beam orientations in radiation therapy: some theoretical considerations *Phys. Med. Biol.* **38** 291–304

[10] Bortfeld T, Schlegel W and Rhein B 1993 Decomposition of pencil beam kernels for fast dose calculations in three–dimensional treatment planning *Med. Phys.* **20** 311–8

[11] Boyer A L, Wackwitz R and Mok E C 1998 A comparison of the speeds of three convolution algorithms *Med. Phys.* **15** 224–7

[12] Brahme A 1995 Treatment optimization using physical and radiobiological objective functions *Radiation Therapy Physics* ed A R Smith (Berlin: Springer) pp 209–46

[13] Burkard R E, Leitner H, Rudolf R, Siegl T and Tabbert E 1995 Discrete optimization models for treatment planning in radiation therapy *Science and Technology for medicine: Biomedical engineering in Graz* ed H Hutten (Berlin: Pabst Science Publishers) pp 237–49

[14] Censor Y, Altschuler M D and Powlis W D 1988 On the use of Cimmino's simultaneous projection method for computing a solution of the inverse problem in radiation therapy treatment planning *Inverse Probl.* **4** 607–23

[15] Chriss T B, Herman M G and Wharam M D 1995 Rapid optimalization of stereotactic radiosurgery using constrained matrix inversion *2nd Congress of the International Stereotactic Radiosurgery Society* (Boston)

[16] Cooper R E 1978 A gradient method of optimizing external–beam radiotherapy treatment plans *Radiology* **128** 235–43

[17] Cotrutz C, Lahanas M, Kappas C and Baltas D 2001 A multiobjective gradient–based dose optimization algorithm for external beam conformal radiotherapy *Phys. Med. Biol.* **46** 2161–75

[18] van Dalen S, Keizer M, Huizenga H and Storchi P R M 2000 Optimization of multileaf collimator settings for radiotherapy treatment planning *Phys. Med. Biol.* **45** 3615–25

[19] van Dyk J, Barnett R D and Battista J J 1999 Computerized radiation treatment planning systems *The modern technology of radiation oncology* ed J. van Dyk (Madison: Medical Physics Publishing) pp 231–86

[20] Engel K 2002 A new algorithm for optimal multileaf collimator field segmentation *Preprint, University of Rostock, Department of Mathematics*

[21] Ezzell G A 1996 Genetic and geometric optimization of three–dimensional radiation therapy treatment planning *Med. Phys.* **23** 293–305

[22] Grossmann C and Terno J 1993 *Numerik der Optimierung* (Stuttgart: Teubner)

[23] Haas O C L 1999 *Radiotherapy treatment planning: new system approaches* (London: Springer)

[24] Hamacher H W and Küfer K-H 2002 Inverse radiation therapy planning – a multiple objective optimization approach *Discrete Appl. Math.* **118** 145–61

[25] Hodes L 1974 Semiautomatic optimization of external beam radiation treatment planning *Radiology* **110** 191–6

[26] Holmes T and Mackie T R 1994 A comparison of three inverse treatment planning algorithms *Phys. Med. Biol.* **39** 91–106

[27] Holmes T W, Mackie T R and Reckwerdt P J 1995 An iterative filtered backprojection inverse treatment planning algorithm for tomotherapy *Int. J. Radiat. Oncol., Biol., Phys.* **32** 1215–25

[28] Hristov D H and Fallone B G 1998 A continuous penalty function method for inverse treatment planning *Med. Phys.* **25** 208–23

[29] Hristov D H and Fallone B G 1997 An active set algorithm for treatment planning optimization *Med. Phys.* **24** 1455–64

[30] Kalinowski 2003 An algorithm for optimal multileaf collimator field segmentation with interleaf collision constraint *Preprint, University of Rostock, Department of Mathematics*

[31] Langer M, Brown R, Morrill S, Lane R and Lee O 1996 A generic genetic algorithm for generating beam weights *Med. Phys.* **23** 965–71

[32] Langer M, Brown R, Urie J, Leong J, Stracher M and Shapiro J 1990 Large–scale optimization of beam–weights under dose–volume restrictions *Int. J. Radiat. Oncol., Biol., Phys.* **18** 887–93

[33] Langer M, Morrill S, Brown R, Urie M, Lee O and Lane R 1996 A comparison of mixed integer programming and fast simulated annealing for optimizing beam weights in radiation therapy *Med. Phys.* **23** 957–64

[34] Lee E K, Fox T and Crocker I 2000 Optimization of radiosurgery treatment planning via mixed integer programming *Med. Phys.* **27** 995–1004

[35] Mackie T R, Scrimger J W and Battista J J 1985 A convolution method of calculating dose for 15–MV X–rays *Med. Phys.* **12** 188–96

[36] Mageras G S and Mohan R 1993 Application of fast simulated annealing to optimization of conformal radiation treatments *Med. Phys.* **20** 639–47

[37] McDonald S C and Rubin P 1977 Optimization of external beam radiation therapy *Int. J. Radiat. Oncol. Biol. Phys.* **2** 307–17

[38] Metcalfe P, Kron T and Hoban P 1997 *The physics of radiotherapy X–rays* (Madison: Medical Physics Publishing)

[39] Morril S M, Lam K S, Lane R G, Langer M and Rosen I I 1995 Very fast simulated reannealing in radiation therapy treatment plan optimization *Int. J. Radiat. Oncol. Biol. Phys.* **31** 179–88

[40] Niemierko A and Goitein M 1990 Random sampling for evaluating treatment plans *Med. Phys.* **17** 753–62

[41] Petti P L, Siddon R L and Bjärngard B E 1986 A multiplicative correction for tissue heterogeneities *Phys. Med. Biol.* **31** 1119–28

[42] Pugachev A, Li J G, Boyer A L, Hancock S L, Le Q-T, Donaldson S S and Xing L 2001 Role of beam orientation optimization in intensity–modulated radiation therapy *Int. J. Radiat. Oncol., Biol., Phys.* **50** 551–60

[43] Pugachev A B, Boyer A L and Xing L 2000 Beam orientation optimization in intensity–modulated radiation treatment planning *Med. Phys.* **27** 1238–45

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

[45] Redpath A T, Vickery B L and Wright D H 1976 A new technique for radiotherapy planning using quadratic programming *Phys. Med. Biol.* **21** 781–91

[46] Rosen I I, Lane R G, Morill S M and Belli J A 1991 Treatment plan optimization using linear programming *Med. Phys.* **18** 141–52

[47] Rowbottom C G, Khoo C S and Webb S W 2001 Simultaneous optimization of beam orientations and beam weights in conformal radiotherapy *Med. Phys.* **28** 1696–702

[48] Shu H Z, Yan Y L, Bao X D, Fu Y and Luo L M 1998 Treatment planning optimization by quasi–Newton and simulated annealing methods for gamma unit treatment system *Phys. Med. Biol.* **43** 2795–805

[49] Söderström A and Brahme A 1992 Selection of suitable beam orientatons in radiation therapy using entropy and Fourier transform measures *Phys. Med. Biol.* **37** 2107–23

[50] Starkschall G 1984 A constrained least–squares optimization method for external beam radiation therapy treatment planning *Med. Phys.* **11** 659–65

[51] Stein J, Mohan R, Wang X-H, Bortfeld T, Wu Q, Preiser K, Ling C C and Schlegel W 1997 Number and orientation of beams in intensity-modulated radiation treatments *Med. Phys.* **24** 149–60

[52] de Wagter C, Colle C O, Fortan L G, van Duyse B B, van den Berge D L and de Neve W J 1998 3D conformal intensity–modulated radiotherapy planning: interactive optimization by constrained matrix inversion *Radiother. Oncol.* **47** 69–76

[53] Webb S 1992 Optimization by simulated annealing of three-dimensional, conformal treatment planning for radiation fields defined by a multileaf collimator: II. Inclusion of two–dimensional modulation of the X–ray intensity *Phys. Med. Biol.* **37** 1689–704

[54] Webb S 1993 *The physics of three–dimensional radiation therapy* (Amsterdam: Institute of Physics)

[55] Wong T P Y, Metcalfe P E and Chan C L 1994 The effects of low–density media on X–ray dose distribution *Medicamundi* **3** 134–43

[56] Wu Q J, Wang J and Sibata C H 2000 Optimization problems in 3D conformal radiation therapy, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **55**, pp 183–94

[57] Wu X and Zhu Y 2001 A global optimization method for three–dimensional conformal radiotherapy treatment planning *Phys. Med. Biol.* **46** 107–19

[58] Wu X and Zhu Y 2001 An optimization method for importance factors and beam weights based on genetic algorithms for radiotherapy treatment planning *Phys. Med. Biol.* **46** 1085–99

[59] Xing L and Chen G T Y 1996 Iterative methods for inverse treatment planning *Phys. Med. Biol.* **41** 2107–23

[60] Yu Y 1997 Multiobjective decision theory for computational optimization in radiation therapy *Med. Phys.* **24** 1445–54