Approximating convex Pareto surfaces in multiobjective radiotherapy planning

David L. Craft,[a] Tarek F. Halabi, Helen A. Shih, and Thomas R. Bortfeld
Department of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School, Boston, Massachusetts 02114

(Received 9 February 2006; revised 7 June 2006; accepted for publication 17 July 2006; published 30 August 2006)

Radiotherapy planning involves inherent tradeoffs: the primary mission, to treat the tumor with a high, uniform dose, is in conflict with normal tissue sparing. We seek to understand these tradeoffs on a case-to-case basis, by computing for each patient a database of Pareto optimal plans. A treatment plan is Pareto optimal if there does not exist another plan which is better in every measurable dimension. The set of all such plans is called the Pareto optimal surface. This article presents an algorithm for computing well distributed points on the (convex) Pareto optimal surface of a multiobjective programming problem. The algorithm is applied to intensity-modulated radiation therapy inverse planning problems, and results of a prostate case and a skull base case are presented, in three and four dimensions, investigating tradeoffs between tumor coverage and critical organ sparing. © *2006 American Association of Physicists in Medicine*.
[DOI: 10.1118/1.2335486]

I. INTRODUCTION

A constant difficulty that arises in radiotherapy treatment planning is the patient-specific tradeoff between dosing the tumor appropriately and keeping the healthy tissue dose low. This is traditionally handled by forming a joint objective function which includes an objective that rewards high, uniform dose to the tumor, as well as separate objectives which penalize dose to various healthy organs. The problem with this approach is that a good set of relative weighting factors on the different objectives is not *a priori* known, and must be found by the treatment planner in a time consuming iterative manner. Furthermore, a set of weights found for one patient will likely not work well for another patient. Indeed, Hunt *et al.*[1] show that for a concave tumor phantom and a nearby critical structure, the geometry parametrized by the separation distance, the relative weights which give comparable plans are different for each instance. The goal of the present article is to present a method which allows treatment planners and physicians to understand the tradeoffs for individual patients, while simultaneously avoiding the time-consuming human-iteration loop of searching for a good set of objective function weights.

Our approach to the multiple objective intensity-modulated radiation therapy (IMRT) optimization problem is to compute a database of Pareto optimal plans for subsequent exploration. For a given set of objective functions $F_i(\boldsymbol{x}), i = 1 \cdots N$, a beamlet solution $\boldsymbol{x}$ is Pareto optimal if there does not exist a strictly better feasible solution. A solution $\boldsymbol{y}$ is strictly better than $\boldsymbol{x}$ if $F_i(\boldsymbol{y}) \leqslant F_i(\boldsymbol{x})$ for all i, with a strict inequality for at least one i. The algorithm we present is suitable for moderate dimensional (N up to 5 or 6) multiobjective problems.

The computation of a representative set of Pareto optimal solutions is not the only approach to multiobjective programming (MOP). We prefer it, however, since it is the only approach which actually shows the tradeoffs to the planners, allowing them to choose a plan with an understanding of the different options available. Two other common approaches for handling multiple objectives are goal programming (GP) (e.g., Ref. 2) and lexicographic ordering (LO) (e.g., Ref. 3). These methods require that the decision maker specifies his or her preferences *before* optimization begins, and based on these preferences, a single Pareto optimal plan is computed, which makes these techniques attractive from a computational standpoint. In GP, objectives are re-expressed as goals. Goals are achieved by minimizing a weighted sum of the deviations from these goals. Based on the set of weights chosen, a single balanced plan is computed. In LO (also called preemptive goal programming), the objectives are prioritized. The first optimization problem minimizes the objective with the highest priority. The result of this serves as a constraint for the next problem, where the objective with the next highest priority is minimized with a constraint that the first objective cannot go above its optimal value in the first solution. This is repeated for subsequent prioritization levels. LO leads to extreme solutions, based on the priority ordering, which may not be desirable by decision makers.

Before reviewing existing algorithms which approximate the Pareto surface (PS), we provide notation for the general Pareto MOP:

$$\begin{aligned} &\text{minimize } \{F_i(\boldsymbol{x})\}, \quad i = 1 \cdots N \\ &\qquad \boldsymbol{x} \in X, \end{aligned} \tag{1}$$

where F_i is the ith objective function, $\boldsymbol{x}$ is a vector of decision variables, and X is a convex constraint set in the decision variable space. Whereas single optimization problems have a single solution, MOPs have a set of optimal solutions called the Pareto set. A version of the above MOP formulation specific to IMRT treatment plan optimization can be found in Ref. 4.

Formulation 1 is shorthand for "find all Pareto optimal solutions with F_i as the objective functions." Typically the

set of all Pareto optimal solutions is not available in closed form. Therefore, one seeks a discrete set of points on the PS which approximates the surface well. (By PS, we mean the surface in objective function space, that is, the space with one coordinate axis for each objective function F_i. One could also speak of the surface of Pareto optimal points in the underlying decision variable $\boldsymbol{x}$ space.)

In settings where the calculation of an individual point on the PS is computationally expensive, there is a motivation to compute a good representation of the surface in as few points as possible. In two dimensions (2D), for example, if the Pareto tradeoff curve is kinked, two end points and a point at the kink produce a sufficient and economical representation of the surface. However, *a priori* the shape of the PS is unknown. Nevertheless, in this article we show how convexity properties of the problem lead to geometric lower and upper bounds on the PS, which guide us to a well-distributed placement of points on the PS.

Scalarization algorithms are a common method for handling problem 1 when the constraint set X is convex and the functions F_i are convex.[5] Under these convexity assumptions, Pareto optimal solutions to problem 1 can be found by solving the following problem:

$$\text{minimize} \sum_{i=1}^{N} w_i F_i(\boldsymbol{x})$$
$$\boldsymbol{x} \in X. \tag{2}$$

By spanning the entire set of weights $w_i \geqslant 0, \sqrt{\sum_{i=1}^{N} w_i^2} = 1$, one obtains the entire PS. However, this is highly impractical, and even a fine discrete sampling is impractical for any number of objectives above 3. Furthermore, there is no guarantee that a prespecified set of weight vectors to run will lead to a good distribution of points on the PS.[6]

Despite these apparent weaknesses, there are several advantages to using a scalarization method. One is that in IMRT planning, many treatment planning systems (TPS) are designed where such scalarization is done by hand, with planners iteratively changing the weights to find a good plan. Therefore, by developing an algorithm which works on this principle, we can hook it up to these solvers (provided they are using convex formulations) with minimal effort. Another advantage is the empirical observation that the solution time for a scalarized problem is much faster than other techniques which get individual points on the PS of a MOP. In particular, the normal boundary intersection (NBI) method,[7] the normal constraint (NC) method,[4,8] and the epsilon constraint method (e.g., Ref. 5) all add hard constraints to the underlying optimization program, and we have observed that these constraints can make the solution process significantly slower. The NBI and NC methods also are designed to populate the PS with an even spread of points regardless of the curvature of the PS, which is not economical. These techniques also clip edge portions off of the PS in dimensions over 2, and thus should be used only when this is acceptable.

The algorithm we present is similar to algorithms presented by Klamroth *et al.*,[9] and Solanki *et al.*,[10] both of which use the idea of sandwiching the PS between a lower and an upper convex approximation. The Klamroth approach is not based on a weighted sum approach, and thus suffers from speed issues for the individual Pareto point solutions. The Solanki approach does use a weighted sum method for generating individual solutions, but there are a couple of key differences with the algorithm we present. One is that we provide a closed form algebraic solution to the problem of finding the distances between the lower approximation and the upper approximation, whereas they rely on a separate linear program for this calculation. The second key difference is in how the boundary of the PS is handled. The Solanki approach uses negative weight vectors, and thus can obtain non-Pareto optimal points. Since each run is expensive, we prefer to not use negative weight vectors, and instead attempt to find the proper weighting vector based on the weight vectors used to create the boundary region we are targeting.

In certain applications, the user will opt to not flush out the boundary points of the Pareto set, in which case the issue of dealing with the boundary is irrelevant. In general though, we assume that the user has formed a set of constraints such that any solution that satisfies those constraints is a potentially viable solution, and hence, we seek to determine the entire PS given that constraint set.

After describing the algorithm, we apply it to two IMRT treatment planning problems, a prostate case and a skull base case. The prostate case demonstrates the algorithm in three dimensions (3D), where we trade off tumor coverage with sparing of the rectum and the femoral heads. Three-dimensional (3D) Pareto surfaces (PSs) can be visualized easily on a computer screen with rotating graphics, and also can be sliced to yield individual two-dimensional (2D) tradeoff curves. For the skull base tumor, we compute a four-dimensional (4D) tradeoff surface. We visualize this in 2D by slicing through two of the four dimensions.

II. METHODS

The algorithm we present relies heavily on the following result from multiobjective programming: if the individual functions F_i are convex and the feasible decision variable space X is convex, then the PS is convex (meaning that any hyperplane plane tangent to the PS is a supporting hyperplane: the PS is entirely on one side of the hyperplane).[11] The method presented in this article is applicable to general convex optimization MOPs.

Vectors are set in bold face. In the objective function space, uppercase letters are used for location vectors (e.g., $\boldsymbol{P}, \boldsymbol{F}$) and lowercase letters are used for direction vectors (e.g., $\boldsymbol{n}, \boldsymbol{w}$).

We begin with a broad description of some of the key geometric ideas of the algorithm, followed with detailed descriptions of each step. At the highest level, the algorithm (called PGEN) iteratively chooses weight vectors $\boldsymbol{w}$ to run to gradually build up the PS. By examining the current points that have been found on the PS along with the weights used in formulation 2 that produced those points, PGEN produces a

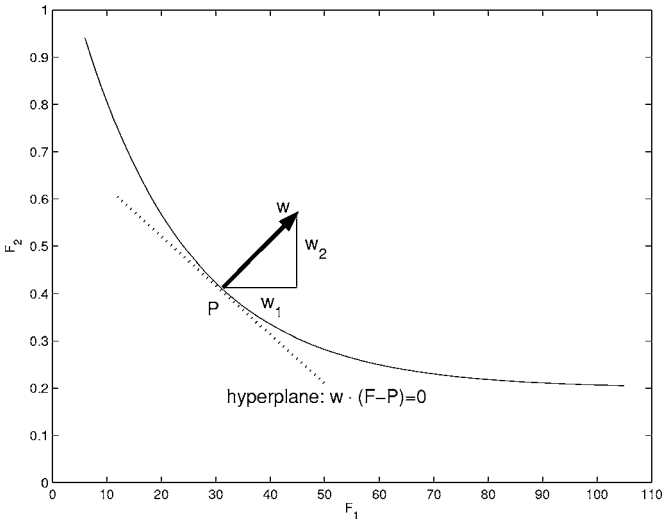

FIG. 1. The hyperplane through the point $\boldsymbol{P}$ with normal $\boldsymbol{w}$, which is given by the equation $\boldsymbol{w}\cdot(\boldsymbol{F}-\boldsymbol{P})=0$ where $\boldsymbol{F}=(F_1,F_2)$, is a lower bound to the PS.

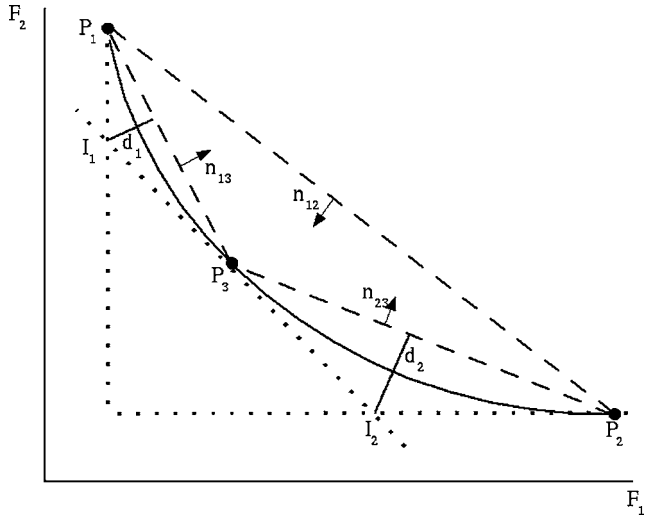

FIG. 2. Illustration of the PGEN algorithm on a bicriteria optimization problem. Intersection of the lower bound tangent planes leads to the points I_1 and I_2, which are called LDP. The distances d_1 and d_2 from these points to the corresponding upper approximation segments are used to select the next $\boldsymbol{w}$ vector to run.

new weight vector to run, and repeats this process until a geometric stop tolerance is met.

Points on the PS and the associated weight vector are written as the pair $(\boldsymbol{P},\boldsymbol{w})$. Since the PS is convex, for each $(\boldsymbol{P},\boldsymbol{w})$, the hyperplane through point $\boldsymbol{P}$ with normal $\boldsymbol{w}$ is a supporting hyperplane (i.e., a lower bound) of the PS. This gives us, with each new point calculated, a new lower bound on the PS. This is illustrated in Fig. 1. An upper bound is obtained by the convex combination of all of the points thus far found on the PS. The lower and upper bounds are then combined, and at places where the bounds have the largest gap, we add new points to the surface.

A. The algorithm

Figure 2 is used to describe various aspects of the algorithm, as visualized in 2D. The concepts extend to N dimensions, but nontrivially, particulary when dealing with the boundary of the PS. We assume for simplicity that each function is to be minimized with positive weights. Maximization problems are turned into minimization problems with negative weights, but this detail is unnecessary and only complicates the exposition.

1. Step 1: Anchor points and a single interior point

Minimizing each objective function individually (by setting $\boldsymbol{w}_i=1$ for the objective function to minimize, and $\boldsymbol{w}_j=0$ for all other objective functions) produces the N anchor points. In Fig. 2, these are the points labeled $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. These points bound the lower extent of the PS in each respective dimension. After the N anchor points are computed, a single balanced point is computed by either using equal weights on each objective function, or by using the outward-facing normal of the hyperplane through the N anchor points, provided all of the components are non-negative.

Refering to Fig. 2, at this point we have calculated points $\boldsymbol{P}_1,\boldsymbol{P}_2$, and an interior point, $\boldsymbol{P}_3$.

2. Step 2: Convex hulling to produce the upper PS approximation

The convex hull of a set of points in N-dimensional space (ND) is the set of all points that can be expressed as convex combinations of the points in the original set. This infinite set is compactly represented as a set of bounding facets (line segments in 2D, triangles in 3D, etc.) and their normals. Since it is known that for convex optimization MOPs the PS points lie on the boundary of a convex set, we can use a convex hulling algorithm to obtain a facet representation of our PS. We use a freely available ND convex hull algorithm (which we will call convhulln) (see Ref. 12), which takes as input the points and returns the connectivity of those points, i.e., the facets of the convex hull, and the face normals. In Fig. 2, convhulln applied to points $\boldsymbol{P}_1,\boldsymbol{P}_2$, and $\boldsymbol{P}_3$ returns the connectivity indicated by the dashed lines. Only the lower facets are used as the upper approximation to the PS, so in this case facets $[\boldsymbol{P}_1,\boldsymbol{P}_3]$ and $[\boldsymbol{P}_3,\boldsymbol{P}_2]$ are used, and facet $[\boldsymbol{P}_1,\boldsymbol{P}_2]$ is thrown away.

Facet rejection in this 2D example is straightforward and could be done by looking at the normal vector of each facet. Using the convention that normals point into the convex hull polygon, we could simply remove the facet whose normal consists of all negative components. In higher dimensions, this simple approach does not work, and whether a facet of the convhulln output should be kept or removed is unclear. For example, consider the 3D PS shown in Fig. 3: any facet consisting of anchor points and edge points, say along the front edge, will have a mixed normal—some positive components and some negative.

In dimensions over 2, it is convenient to think of facets belonging to one of three categories. The first category are facets which have strictly positive components for their normal vector. These facets are the true upper approximation facets to the PS: all of the points belonging to such a facet

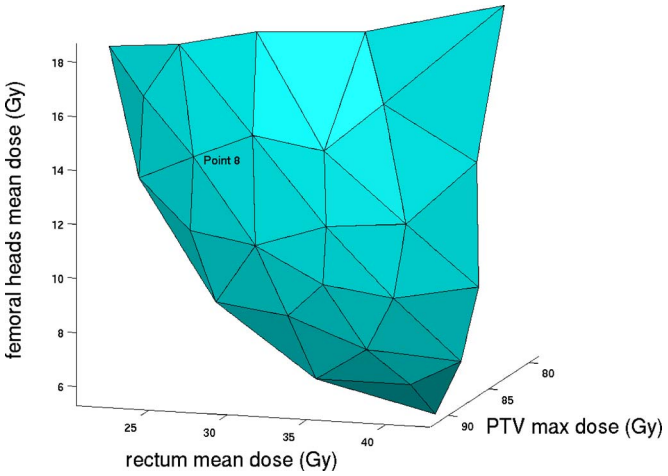

FIG. 3. 3D PS for a prostate case. The vertices of the triangular facets are the computed Pareto optimal points. The vertex labeled point 8 is selected as a good compromise plan, and its dose distribution is shown in Fig. 5.

are not dominated by any point in the current Pareto set (the current vertices). The second category are facets with some positive and some negative normal vector components. In these facets, the vertices are Pareto optimal, but each interior point is dominated by one of the vertices of that facet. While such a facet might not make sense as an upper approximation, it can still be a useful *generating facet*, a facet which may generate further PS points. Third, there are facets with all negative normal components. The facet connecting the anchor points is frequently of this type. Facets of this type are not useful for generating new points and are discarded.

Even with this categorization, many facets during the course of the algorithm will be in a gray zone regarding their usefulness for PS point generation. Rather than a fuzzy evaluation based on looking at how many normal components are negative, how negative are they, etc., we base our acceptance/rejection on whether or not we can intersect the lower bound vertex hyperplanes, as described next.

3. Step 3: Calculating the lower distal points

Recall that hyperplanes through the PS points, with normals vectors equal to the weight vector $\boldsymbol{w}$ yielding that point, are lower bounds of the PS. To find the maximal distance between the upper facets and the associated lower bounds, we first calculate the position of the lower distal point (LDP) associated with each facet that has not already been discarded due to an all negative normal vector (the facet $[\boldsymbol{P}_1,\boldsymbol{P}_2]$ in Fig. 2 has an all negative normal vector, and so a LDP is not calculated for it). The LDPs in Fig. 2 are the points $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ ("$\boldsymbol{I}$" for intersection). For each facet, the LDP is calculated by intersecting the lower bound hyperplanes at each vertex, provided these N hyperplanes intersect in a unique point. The lower bound hyperplane at a point $\boldsymbol{P}$ with normal $\boldsymbol{n}$ is given by $\boldsymbol{n}\cdot(\boldsymbol{F}-\boldsymbol{P})=0$. Solving N of these equations, from the N vertices of a facet, for the vector $\boldsymbol{F}$, produces the LDP for that facet. If the N hyperplanes do not intersect, or intersect nonuniquely, the facet is handled as a boundary facet, as described below. Ideally, the location of the LDP will be checked to make sure it is within the bounding box of the PS since, at edge facets in particular, the hyperplane intersection LDP may be far from the actual PS, and should therefore be discarded. The difficulty is, while the anchor points give us the lower extent of the PS, it is nontrivial to obtain the maximal extent in each dimension.[10] Often though, the upper extent for each dimension will be simply the maximum coordinate of the points already found (empirical observation).

4. Step 4: Calculating the distances from the upper approximation facets to the LDPs

This step calculates the distance from each upper bound facet (e.g., $[\boldsymbol{P}_1,\boldsymbol{P}_3]$) to its associated LDP (e.g. $\boldsymbol{I}_1$). The following formula gives the distance d from a point $\boldsymbol{I}$ (the LDP from the previous step) to a hyperplane with normal $\boldsymbol{n}$ given by the equation $\boldsymbol{n}\cdot\boldsymbol{F}=c$:

$$d=\mathrm{abs}[(\boldsymbol{n}\cdot\boldsymbol{I}-c)/\|\boldsymbol{n}\|]. \tag{3}$$

$\|\cdot\|$ denotes the Euclidean norm of the vector, and abs denotes absolute value. To use Eq. (3) we require the normal $\boldsymbol{n}$ of a hyperplane defined by N vertices, which is available from the output of convhulln. If not available (if convhulln is called through MATLAB, the normal vectors are not returned), it can be calculated as follows. For N linearly independent points in ND, the normal to their hyperplane is the null space of $N-1$ linearly independent vectors in the hyperplane. Choosing one vertex arbitrarily, and forming the other $N-1$ differences, we calculate the normal as the nullspace of these differences. The constant c in the hyperplane equation is calculated using any of the N vertices $\boldsymbol{V}$: $c=\boldsymbol{n}\cdot\boldsymbol{V}$.

This distance calculation is done for every facet that was not removed in step 2, and which has a LDP in the bounding box of the current PS. If the maximum distance is below the stopping threshold, and the boundary option is turned off, the algorithm is finished at this point.

5. Step 5: Choosing the next facet to run

As long as there are facets with d greater than the stop tolerance, then the facet with the largest d is the next facet to run. Once d is below the tolerance, the algorithm terminates if the user has selected the no boundary option. Otherwise, if there are boundary facets, the one with the smallest angle is chosen to run next. The angle for a facet is defined as the angle between the facet normal vector $\boldsymbol{n}$ and the average of the vertex weight vectors $\bar{\boldsymbol{w}}$. The choice to select the smallest angle facet stems from the observation that for facets flush with the PS, $\boldsymbol{n}$ and $\bar{\boldsymbol{w}}$ are in roughly the same direction, as opposed to say the facet which connects the anchor points, in which $\boldsymbol{n}$ and $\bar{\boldsymbol{w}}$ point in opposite directions, see Figs. 1 and 2.

6. Step 6: Computing the next w and running it

Given the next facet to run, there are two logical choices for the next $\boldsymbol{w}$ to run. One choice is to use the normal of the facet hyperplane for $\boldsymbol{w}$. This method is problematic if the hyperplane normal has negative components. An alternate

method is to use a weighted combination of the vertex weight vectors. Rather than the average of the vertex weight vectors, we use a weight vector which is a convex combination of the vertex $\boldsymbol{w}_i$'s that is maximally different from the vertex $\boldsymbol{w}_i$'s. To see why this choice is sensible over using the average, consider a 3D problem with three vertex $\boldsymbol{w}_i$'s which are linearly dependent. Say $\boldsymbol{w}_3$ is the average of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, then the average of all three vectors will be $\boldsymbol{w}_3$, and we would rather choose a new and maximally different $\boldsymbol{w}$. In practice, a workable approach is to use the hyperplane normal if it is non-negative in all components, otherwise use the vertex maximally different $\boldsymbol{w}$, which is found by simple nonlinear optimization.

Having selected a new $\boldsymbol{w}$, and checking that it has not already been run, we use it in problem 2. This gives a point on the PS, which is added to the existing set of points, and the algorithm returns to step 2. It is possible that the new point has already been found, in which case the facet used to generate the new point is flagged so as not to be run again. The next sections discuss some implementation refinements and additions.

B. Dimension normalization

When the individual objective functions are of different magnitudes, the dimensions should be scaled so that, for example, distance calculations between the LDPs and hyperplanes are not dominated by the large magnitude objective function(s). This is done by mapping each objective value into $[0, 1]$ by using its minimum and maximum value on the PS. The minimum values are the anchor points, calculated in step 1. The maximum values, while not known, can be replaced by the current maximum values attained. Letting L_i and U_i denote the minimum and maximum levels of objective function i, the normalization of a point $\boldsymbol{P}$ on the PS is given by

$$\bar{P}_i = \frac{P_i - L_i}{U_i - L_i}, \quad i = 1 \cdots N. \tag{4}$$

C. Filling in additional solutions post-optimization

After the algorithm terminates, there is a simple way to increase the density of points on the PS that takes advantage of the problem convexity.[13] Consider a facet of the upper approximation consisting of vertices with associated beamlet intensity solutions denoted $\boldsymbol{x}_i$. Let $\boldsymbol{x} = 1/n \Sigma \boldsymbol{x}_i$ be the weighted average beamlet intensity vector, which is a feasible solution. The convexity assumptions on the F_j functions yield that for each objective function j we have $F_j(\boldsymbol{x}) \leqslant 1/n \Sigma F_j(\boldsymbol{x}_i)$. That is, the point in objective function space that we obtain by evaluating the objective functions at an $\boldsymbol{x}$ which is in the center of each facet is at least as good as the linear facet approximation.

The facet approximation is an upper approximation to the PS, exact at the vertex points. With the above observation, we can improve on our upper approximation by evaluating the objective functions at facet centers (or anywhere else on the facet). This can be done to as fine as resolution as desired. While the new points are not guaranteed to be on the PS, they are at least as good as our original facet-based upper approximation.

III. RESULTS

We apply the algorithm to two clinical examples. For the dose calculation, we use the IMRT planning package KonRad[14–18] that was originally developed at the German Cancer Research Center (DKFZ). The prostate case has a voxel size of $2 \times 2.5 \times 2$ mm, for a total of 831 024 voxels. The skull base case has a voxel size of $2.44 \times 2.5 \times 2.44$ mm for a total of 510 300 voxels. To reduce the problem size, a 1% sampling of the unclassified tissues was used for the skull base case, and two shells, one surrounding the prostate and one near the skin surface, were used in the prostate case, instead of the entire unclassified tissue. Both cases use 5×5 mm beamlet sizes, the prostate case having a total of 785 beamlets and the skull base case 1099 beamlets. The optimizations are done using the CPLEX 9.1 barrier method for linear programming with default settings.

The two cases presented herein are modeled with linear objective functions and constraints, and hence, satisfy the prerequisite conditions of the PGEN algorithm: convexity of the objective functions and convexity of the feasible set.

The first case is a prostate tumor with seven equispaced beams, where we examine the 3D between tumor dose homogeneity, rectal sparing, and femoral head sparing. To achieve tumor dose homogeneity, we enforce that every voxel achieves at least 75 Gy, and the tumor objective is to minimize the maximum dose (to see how this is handled in a linear optimization setting, refer to Ref. 4). For both the rectum and the femoral heads, the objective is to minimize the average dose. Additional constraints which penalize doses above a threshold value are used to achieve conformality around the prostate and restrict the entrance doses. Figure 3 shows the full 3D PS, and Fig. 4 shows the same surface by slicing through the planning target volume dimension. This prostate case is meant as a demonstration of a 3D tradeoff, and therefore we have chosen not to include a separate clinical target volume (CTV). A more realistic setting might include the CTV dose, tumor coverage conformality and entrance doses as additional tradeoff objectives.

Figure 4 shows that the tradeoff between average rectum dose and average femoral dose is smooth, as opposed to kinked. The isolevels for the tumor dimension confirm that as we enforce more tumor dose homogeneity, the rectum and femoral heads doses get increasingly worse. The eighth PS point, indicated in Fig. 3, is selected as a good tradeoff solution, and the dose wash is shown for this plan in Fig. 5.

The second clinical example is a skull base tumor case, with five equispaced beams. The patient was a 62 year old gentleman who presented with a recurrent skull base meningioma arising along the right sphenoid sinus and involving the right orbital apex and maxillary sinus. Following a failed attempt of curative surgery, he was treated with definitive radiation and was planned for a target dose of 60 Gy. Fol-

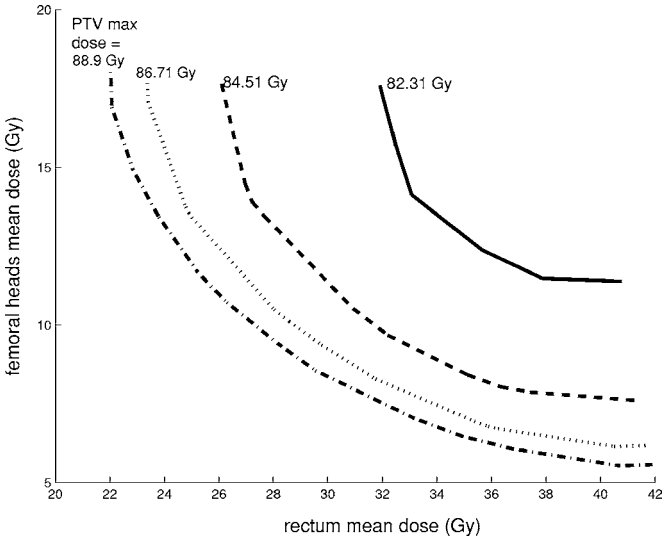

FIG. 4. Slices of the 3D prostate case PS in Fig. 3. The solid line represents the highest level of tumor dose homogeneity, and hence, the values for the rectum mean dose and femoral heads mean dose are relatively high, due to the difficulty of achieving high tumor dose homogeneity.

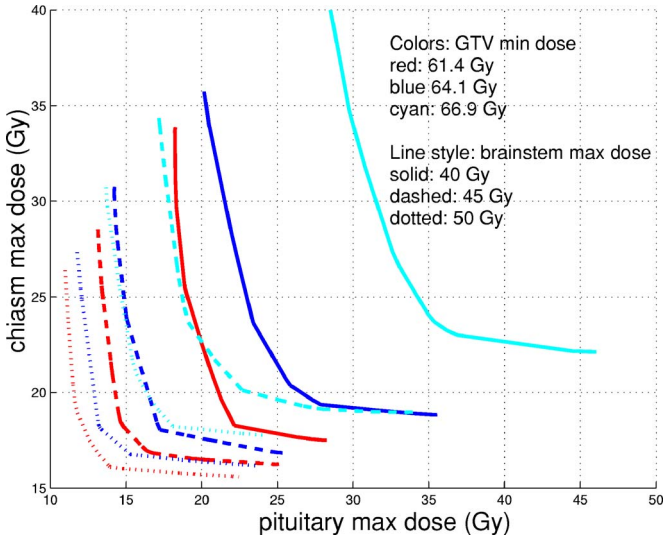

FIG. 6. A 4D PS for a skull base case. The line style is for different brainstem maximum dose levels: solid 40 Gy, dashed 45 Gy, and dotted 50 Gy. The set of solid, dashed, and dotted curves closest to the origin are for GTV minimum dose of 61.4 Gy. The next set outward are for 64.1 Gy, and the upper rightmost are for 66.9 Gy.

lowing a standard allowance for limited hotspots of approximately 10%–15%, we accepted an upper dose limit of 70 Gy. Doses lower than 60 Gy were deemed not acceptable and would compromise probability of tumor control. Doses as high as 70 Gy were concerning for osteoradionecrosis but if were limited in volume or maintained within the tumor were felt to be acceptable.

Dose constraints applied to surrounding normal structures were selected to minimize radiation-induced toxicity. Brainstem dose was limited to 55 Gy. Pituitary constraints are often not used thus we chose 70 Gy to allow for maximal range of solutions and appreciation of dose tradeoffs between structures.

For this case, we compute a 4D PS using the following four objectives: minimize the brainstem, pituitary, and chiasm maximum dose levels, and maximize the minimum tumor GTV dose (while the maximum tumor dose, GTV and CTV, is capped at 70 Gy). The resulting PS is shown in Fig. 6, from a database consisting of 266 plans. This relatively large number of plans was chosen to explicitly show the shape of the PS, but characteristics of the tradeoffs in this example are available in a database with as few as 63 plans (figure not shown). This figure offers a complete view of the 4D PS in a single 2D graph, thus a complete picture of the tradeoffs involved for this case. The planner might look at such a figure, initially considering a plan at the kink of the dotted red curve in Fig. 6, with the DVH shown in Fig. 7 but then, given the improvement to both tumor coverage and brainstem sparing that is achievable with tolerable increases to the pituitary and chiasm, he/she may decide on the plan shown by the DVH in Fig. 8. Dose washes for this plan are shown in Figs. 9 and 10. The full knowledge of these tradeoffs at decision time is what makes the PS approach so valuable.

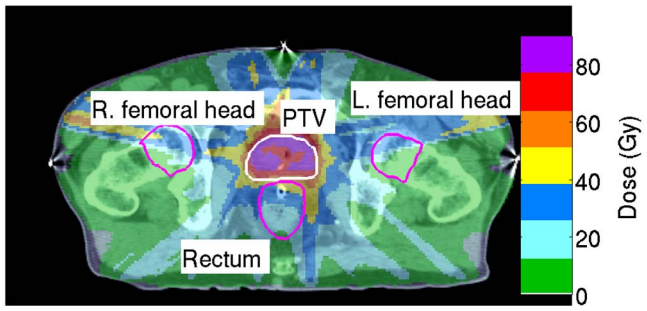

FIG. 5. Dose wash for PS point 8, indicated in Fig. 3. This solution emphasizes rectal sparing. The maximum tumor dose is 86.6 Gy, the mean rectum dose is 25 Gy, and the femoral head mean dose is 13.5 Gy.

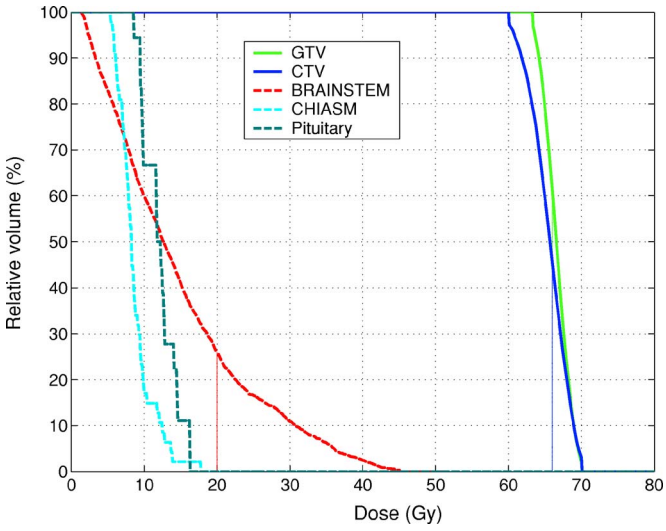

FIG. 7. Dose volume histogram for solution 5 from Pareto plan database for skull base case. This is an equibalanced solution.

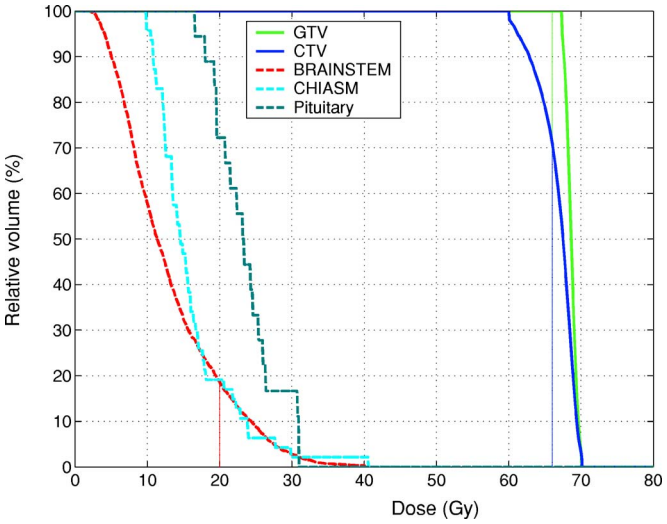

FIG. 8. Dose volume histogram for solution 63 from Pareto plan database for skull base case. This solution is more aggressive regarding boosting the GTV min dose and pushing down the brainstem max dose. The result is a plan that compromises the chiasm and pituitary more.

IV. DISCUSSION AND CONCLUSION

We present an algorithm for calculating well-placed points on a PS, and apply it to the IMRT planning problem. The algorithm requires that the feasible set and objective functions are convex, and in particular we satisfy this by staying within a linear programming environment. The sandwiching technique (i.e., finding upper and lower bounds for the surface) allows the PS to be iteratively approximated to any specified degree of exactness: the algorithm specifies how to add a single point to the surface to best reduce the surface position uncertainty. These benefits are not available in the normal constraint method, the normal boundary intersection method, the epsilon constraint method, or the uniform weights scalarization approach.

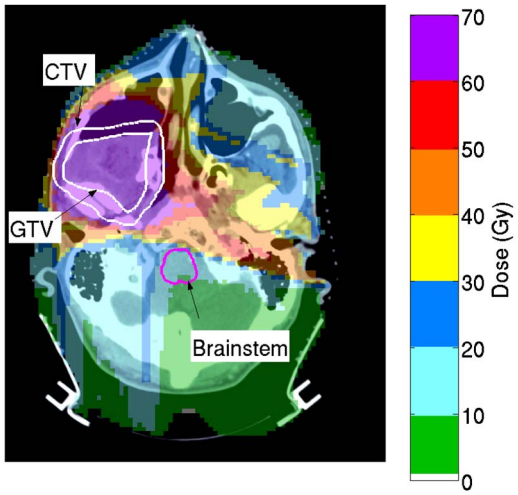

FIG. 9. Dose wash for solution 63 from Pareto plan database for skull base case. Axial view, showing tumor coverage.

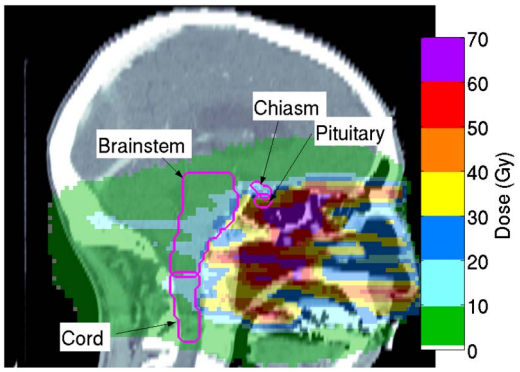

FIG. 10. Dose wash for solution 63 from Pareto plan database for skull base case. Sagittal view, showing avoidance of critical structures (this slice does not contain any tumor voxels, but the adjacent slice does, hence the presence of high dose voxels).

Our choice of linear programming as well as the technique of convex hulling to represent and sandwich the PS both warrant discussion. The benefits of linear programming formulations are that they can be solved to optimality (although without special implementations this can be time intensive, see below), and hard constraints can be imposed. Hard constraints can be used to reduce the dimensionality of the Pareto tradeoff by modeling some organs with constraints instead of objectives.

On the negative side, hard constraints (e.g., minimum and maximum tumor doses) can lead to infeasibility, which then requires human intervention time which we are trying to eliminate in the first place. A typical single run for the cases presented herein takes between 10 and 30 min with a Pentium 4 with 2 GB of random access memory. Assuming two runs per hour, this means 30 solutions over a 15 h period (overnight), which is enough to populate a 3D surface but probably a little sparse for a 4D surface. Linear optimization software which has been customized to IMRT, e.g., Refs. 19–21, should make the computation of higher dimensional PS (up to $\approx$ 5 or 6 dimensions) feasible with this approach.

The use of convex hulling to triangulate the current points on the surface also places a similar restriction of the dimensionality of the problem, since convex hulling time and the number of facets goes up exponentially with dimension[22] (see also Ref. 23). At this point, however, we do not see a major drawback to restricting the number of objectives to fewer than six since currently, tradeoff surfaces are not used at all in treatment planning, and so showing tradeoffs even in 2–5 dimensions will be a useful addition. Furthermore, many clinical IMRT cases have important tradeoffs amongst only a few volumes-of-interest (VOIs). If the number of VOIs is truly greater than 5, and it is not known which ones offer the most clinically relevant tradeoffs, a hybrid LO/PS generation approach could consist of an initial set of runs to find nontrivial bounds for some of the VOIs, with these bounds then serving as constraints for the subsequent PS generation.

An alternate PS generation approach[13] being worked on concurrently at the ITWM Center for Industrial Mathematics in Germany uses nonlinear programming and avoids convex

hulling, thus making it scale into higher tradeoff dimensions more easily. In the linear programming setting, one can trade off two objectives and put hard constraints on all the others, and therefore produce a convex tradeoff PS for visual inspection. One downside of the nonlinear approach is the difficulty of trading off a small number of objectives, say two, while constraining the other structures for clinical feasibility. Since constraints are handled by a penalty method in most large-scale nonlinear solvers, constraints are put into the objectives with high weights, while smaller weights are put on the normal objectives. As the weights are changed to populate the PS the *relative* weights between the objectives and the penalized constraints are changing. This causes each calculated point to be on a different slice of the expanded PS, meaning the PS which includes the constraints as penalty objectives. When such points are projected to the original (nonexpanded) tradeoff space, they will not fall on a convex surface. To see this, imagine projecting points from a 3D convex pareto surface onto two of the dimensions: the resulting points do not fall on a convex 2D curve. While still potentially practical, this makes nonlinear approaches difficult to visualize. Due to the infancy of multicriteria optimization in IMRT planning, the visualization of tradeoff surfaces seems especially important.

Similar to the approach adopted in Ref. 13, we adopt a scalarization approach for the MCO. In addition to the two advantages already discussed, good individual run speed and the fact that current single objective TPS systems use scalarization approaches, the scalarization approach naturally finds kinks in the PS, as seen in the geometry of the approach, see Fig. 2. If point $\boldsymbol{P}_3$ was at a kink, then a large range of $\boldsymbol{w}$'s would detect it, thus making it an early point found. This is a major benefit for approximating the PS with few points.

Future work in this area will be guided largely by clinical experience and physician feedback. However, some technical issues are already evident. Incorporating dose-volume (DV) constraints and objectives is a computational challenge, but highly valuable due to their clinical popularity. How to include DV constraints into an optimization formulation is a subject of much research, see, e.g., Refs. 24–26, since they make the underlying feasible space nonconvex.[27] However, the practical consequences of the added nonconvexity may be minimal,[26] and it may be that additional constraints, such as normal tissue constraints, make the feasible set convex even with DV constraints. Currently, promising ideas for dealing with DVH requirements in the linear framework involve approximating the DVH requirements in such a way as to preserve convexity.[20,28,29]

The motivation for computing a database of Pareto optimal plans comes from the observation that planners spend a lot of time tweaking input parameters until a suitable plan is found.[30] In forming a PS of plans, we automatically run a set of input parameters, avoiding the human iteration loop. PSs by definition, however, are formed by adjusting the relative weights between objective functions, but in practice planners adjust internal function parameters as well. For example, a planner using a one-sided quadratic penalty term for an OAR has the freedom to adjust the weight on the objective function, the power (2 for quadratic, 3 for cubic, etc.), and the threshold dose below which a voxel's contribution is 0. New approaches which address how to automatically adjusts function parameters in addition to weights to produce a database of useful solutions may prove very useful to the field. The navigation of such mixed databases, as well as the pure PSs presented herein, is a nontrivial task. The challenge is to find a technology that allows the complexity of a Pareto database to be probed in an intuitive yet nonrestrictive manner.

ACKNOWLEDGMENTS

The authors are grateful to Dr. Karl-Heinz Küfer, Dr. Michael Monz, and Dr. Alexander Scherrer from Fraunhofer ITWM, Kaiserslautern for helpful discussions of this work, as well as Tim Chan, Ben Martin, and Alex Trofimov at Massachusetts General Hospital. This work was supported by NCI Grant No. 1 R01 CA103904-01A1: Multi-criteria IMRT Optimization.

[a]Electronic mail: dcraft@partners.org

[1]M. A. Hunt, C. Y. Hsiung, S. V. Spirou, C. S. Chui, H. I. Amols, and C. C. Ling, "Evaluation of concave dose distributions created using an inverse planning system," Int. J. Radiat. Oncol., Biol., Phys. **54**, 953–962 (2002).
[2]J. P. Ignizio, *Goal Programming and Extensions* (Lexington Books, Lexington, MA, 1976).
[3]K. Miettinen, *Nonlinear Multiobjective Optimization* (Kluwer, Dordrecht, 2004).
[4]D. Craft, T. Halabi, and T. Bortfeld, "Exploration of tradeoffs in intensity-modulated radiotherapy," Phys. Med. Biol. **50**, 5857–5868 (2005).
[5]S. Ruzika and M. M. Wiecek, "A survey of approximation methods in multiobjective programming," Technical report, Department of Mathematics, University of Kaiserslautern, 2003.
[6]I. Das and J. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," Technical report, Dept. Of Computational and Applied Mathematics, Rice University, 1996.
[7]I. Das, "An improved technique for choosing parameters for pareto surface generation using normal-boundary intersection," *WCSMO-3 proceedings*, 1999.
[8]A. Messac, A. Ismail-Yahaya, and C. A. Mattson, "The normalized normal constraint method for generating the pareto frontier," **25**, 86–98 (2003).
[9]K. Klamroth, J. Tind, and M. M. Wiecek, "Unbiased approximation in multicriteria optimization," Math. Methods Oper. Res. **56**, 413–437 (2002).
[10]R. S. Solanki, P. A. Appino, and J. L. Cohon, "Approximating the noninferior set in multiobjective linear programming problems," Eur. J. Oper. Res. **68**, 356–373 (1993).
[11]M. Ehrgott, *Multicriteria Optimization*, 2nd ed. (Springer, Berlin, 2005).
[12]www.qhull.org
[13]K.-H. Küfer, A. Scherrer, M. Monz, F. Alonso, H. Trinkaus, T. Bortfeld, and C. Thieke, "Intensity modulated radiotherapy—A large scale multicriteria programming problem," OR Spectrum **25**, 223–249 (2003).
[14]T. Bortfeld, J. Stein, and K. Preiser, "Clinically relevant intensity modulation optimization using physical criteria," in *XIIth International Conference on the Use of Computers in Radiation Therapy*, edited by D. D. Leavitt and G. Starkschall (Medical Physics, Madison, 1997), pp. 1–4.
[15]K. Preiser, T. Bortfeld, K. Hartwig, W. Schlegel, and J. Stein, "A new program for inverse radiotherapy planning," in *XIIth International Conference on the Use of Computers in Radiation Therapy*, edited by D. D. Leavitt and G. Starkschall (Medical Physics, Madison, 1997), pp. 425–428.
[16]U. Oelfke and T. Bortfeld, "Inverse planning for photon and proton beams," Med. Dosim **26**, 113–124 (2001).
[17]S. Nill, "Development and application of a multi-modality inverse treatment planning system," P.h.D. thesis, University Heidelberg, 2001.

[18]T. Bortfeld, J. Burkelbach, R. Boesecke, and W. Schlegel, "Methods of image reconstruction from projections applied to conformation radiotherapy," Phys. Med. Biol. **35**, 1423–1434 (1990).
[19]E. Lee, T. Fox, and I. Crocker, "Integer programming applied to intensity-modulated radiation treatment planning," Ann. Operat. Res. **119**, 165–181 (2003).
[20]H. E. Romeijn, R. K. Ahuja, J. F. Dempsey, A. Kumar, and J. G. Li, "A novel linear programming approach to fluence map optimization for intensity modulated radiation therapy treatment planning," Phys. Med. Biol. **48**, 3521–3542 (2003).
[21]A. Scherrer, K-H. Küfer, T. Bortfeld, M. Monz, and F. Alonso, "IMRT planning on adaptive volume structures—A decisive reduction in computational complexity," Phys. Med. Biol. **50**, 2033–2053 (2005).
[22]C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, "The quickhull algorithm for convex hulls," ACM Trans. Math. Softw. **22**, 469–483 (1996).
[23]mathworld.wolfram.com/convexhull.html
[24]M. C. Ferris, R. Einarsson, Z. Jiang, and D. Shepard, "Sampling issues for optimization in radiotherapy," Technical report, Computer Sciences Department, University of Wisconsin, 2004.
[25]M. C. Ferris, J. Lim, and D. Shepard, "Radiosurgery treatment planning via nonlinear programming," Ann. Operat. Res. **119**, 247–260 (2003).
[26]J. Llacer, J. Deasy, T. Bortfeld, T. Solberg, and C. Promberger, "Absence of multiple local minima effects in intensity modulated optimization with dose-volume constraints," Phys. Med. Biol. **48**, 183–210 (2003).
[27]J. O. Deasy, "Multiple local minima in radiotherapy optimization problems with dose-volume constraints," Med. Phys. **24**, 1157–1161 (1997).
[28]M. Chu, Y. Zinchenko, S. G. Henderson, and M. B. Sharpe, "Robust optimization for intensity modulated radiation therapy treatment planning under uncertainty," Phys. Med. Biol. **50**, 5463–5477 (2005).
[29]T. Halabi, D. Craft, and T. Bortfeld, "Dose-volume objectives in multi-criteria optimization," Technical report, Phys. Med. Biol. **51**, 3809–3818 (2005).
[30]C. Thieke, K-H. Küfer, M. Monz, A. Scherrer, F. Alonso, S. Nill, C. Thilmann, and T. R. Bortfeld, "Beyond weight factors: New concepts for defining and analysing dose optimisation," Radiother. Oncol. **73**, s-75 (2004).