Design Optimization Using Dynamic Evaluation

Witold Paluszynski
University of Washington
Computer Science Department, FR-35
Seattle, WA 98195

Ira Kalet, Ph.D.*
University of Washington
Radiation Oncology Department, RC-08
Seattle, WA 98195

Abstract

We describe a search strategy that may be useful for a class of design problems by developing an example from cancer radiation treatment planning. This application problem involves typical features of design problems such as constraints, optimality, a large search space with continuously varying parameters as well as discrete (non-numeric) parameters. There is no known method of comparing elements of the solution space based on a static evaluation function. We have therefore developed a *dynamic evaluation function,* which attempts to heuristically compare all solutions with one another, as a way of interpreting the evaluation results. This allows us to use an analog of hill-climbing with a simple SELECT-GENERATE-TEST loop where expert rules are used as "move generators'' and a similarity metric is used to control or direct the application of the rules for plan modification. Preliminary tests of these ideas indicate that a practical working system can be built.

1 Problem definition

Design problems have received a lot of attention recently in AI research [Mostow, 1985]. Typically, design tasks present difficult problems with big search spaces and solutions defined in terms of continuously varying parameters. They usually involve constraints and optimality criteria. Analytic solutions generally do not exist and experiential "rules of thumb" are not sufficient. This is because it is often necessary to reason about complex properties of objects, such as their geometry, and incrementally approach the best solution by drawing conclusions from the explored variants. The few existing systems such as: AIR/CYL [Brown and Chandrasekaran, 1986], PRIDE [Mittal and Araya, 1986], VT [Marcus et a/., 1988], or BTExpert [Adeli and Balasubramanyam, 1988], present ad hoe or partial solutions to problems rather than a systematic methodology.

The problem domain we are addressing is radiation therapy planning. Treatment of cancer with radiation usually involves setting up several radiation sources to fit the patient's geometrical shape and medical status. The objective is to achieve a high, relatively uniform dose to the disease area (tumor), while keeping normal tissue dose within tolerance constraints [Bleehen *et a/.*, 1983]. An expert radiation therapy planner (called a dosimetrist) uses graphic simulation programs (called treatment planning programs, see [Kalet and Jacky, 1982]) to obt ain feedback on tentative designs that may need considerable refinement. The output of the simulation is a set of radiation dose contour plots that are difficult to interpret except by the trained eye of the dosimetrist.

We have previously reported on an expert system that makes the high-level decisions of radiation therapy planning using production rules and prototypes as plan building blocks [Kalet and Paluszynski, 1985], [Paluszyriski and Kalet, 1987]. The next step was to use the knowledge of the physical properties of radiation and the biological knowledge of the tissue response to radiation to make sure that the constructed treatment plan is acceptable and optimal in some sense [Paluszyriski, 1989]. This is done by identifying defects in the plan and trying to fix them by modifying some of the beams. Because of the specific problem with comparative evaluation of candidate designs (see section 2 below) this step can only be effective to a limited extent. At some point the knowledge-based "plan optimizer" creates a lot of possible plan improvement suggestions and cannot suggest which one offers the best chance of success. In sections 3 and 4 below we demonstrate a technique of augmenting such a knowledge-based reasoning system to make the optimization cycle more effective and easier to control. We believe this technique will solve this problem and perhaps other similarly ill-conditioned ones.

It is difficult to create simplified or abstract problem cases that will exercise the ideas we have developed. There is no way to verify anything with a stripped-down system which generates trivial plans. Also, there is no possibility of building a small scale prototype — the system has to provide answers at least for some class of tumors. This requires the full capabilities of the simulation and all the details of sophisticated radiation treatment. For example, we have had to incorporate the handling of

*This work was partially supported by NIH grant no. LM04174 from the National Library of Medicine and contract no. CM97282 from the National Cancer Institute.

the geometry of shielding lead blocks, wedge filters and so on. Without this, the knowledge of plan improvement will rarely apply. We have not implemented the complete system yet but have tested the ideas presented on some real patient cases. The preliminary results show that the method is promising enough to warrant implementing a full prototype system [Paluszynski, 1989].

2 Evaluating candidate solutions

In most problems where artificial intelligence techniques have been applied to search for a solution there is a way of evaluating a candidate solution as soon as, or even before, it is fully constructed. In other words, just by looking at a candidate one can determine how good it is. Such an evaluation mechanism is called a *static* evaluation function. Further, one can compare the value results of two candidates to determine which one is "better." An evaluation function which provides such capability will be called *scalar.* As we will demonstrate here, constructing an evaluation function for radiation therapy design which would be both static and scalar appears to be impossible. This unfortunate fact is the main motivation for the approach presented here.

2.1 Static evaluation

A measure of "goodness" of a radiation treatment plan has to include the distribution of radiation doses in the cancerous target volume as well as within the normal tissue and especially critical organs which are particularly sensitive to radiation. The evaluation principle can be stated as follows:

> A plan which delivers doses within *±5%* of the prescribed tumor dose to all defined tumor area, and keeps the doses within all critical organs below their respective tolerance doses has the maximum chance of curing the patient.

This principle defines the constraints on the treatment plans but in a difficult realistie case it may be next to impossible to find a plan satisfying them. If an ideal situation cannot be achieved the physician may accept a slight underdose of the tumor (up to 10%), a narrower safety margin around it, or even overdosing a critical organ up to the point of sacrificing the organ. The mechanism of making those subtle concessions seems to be impossible to model using any mathematical function combining solution features.

A great deal of research has been done on trying to construct such an evaluation function [Wolbarst *et al.*, 1982]. It is desirable for two main purposes: (i) for objective evaluation of treatment plans, for example in order to decide whether to accept one or not, and (ii) for subjective evaluation, for example to decide which of two candidate plans is a better material for further optimization. It is interesting to note that these purposes are partially contradictory, at least as far as the radiation therapy design is concerned. Objective evaluation requires more elaborate results to be useful. For example, it is hard to tell what needs improvement in a plan with the evaluation result of 0.6. But this type of simple and concise evaluation results are ideal for the subjective (comparative) evaluation.

2.2 Dynamic evaluation

This led us to abandon the search for an evaluation scheme which would be both static and scalar. Instead, we developed two evaluation mechanisms and the system uses the results which it currently needs. The first phase is static evaluation which, after analyzing the simulated dose distribution within the patient's body, produces a list of *trouble spots.* These are locations of doses being in violation with the evaluation principle stated above. They appear as either cold spots or hots pots, depending of the direction of this violation. These results are used directly by a knowledge-based system which attempts to "fix" any defects in a treatment plan.

The second evaluation phase, called *dynamic evaluation,* uses the results of the first phase. For any two given plans, it tries to determine which one is better. Such a question can only occasionally be answered in a categorical way (otherwise evaluation of plans would not be so hard). Therefore we introduce some heuristics to guess which plan is likely better and we also allow this question to remain unanswered. In other words, the evaluation function described here will be partial. This is unavoidable but, since this function determines the directions of search, having too little information we may end up exploring many unnecessary candidates. However, the search technique presented below (section 4) also works with incomplete evaluation results. When the information is not available for a given point in the solution space it looks in the immediate neighborhood. The more information is available the more efficiently can it be used.

The actual form of the evaluation results is a collection of vectors in the solution space. Plans are compared pairwise with respect to several heuristic measures of "plan goodness" and the results of these comparisons are stored in each plan as unit length vectors which point toward the "better" plan. Any treatment plan can have a number of such evaluation vectors associated with *it.* Plans having a lot of inward pointing vectors are those which compare favorably with a lot of other plans, and vice versa. Aside from such clearly "successful" or "failed" plans the vectors carry much more information in them. For example, a plan which has a few of both inward and outward vectors can be identified as being on a successful optimization path if a lot of incoming vectors come from one general direction and a lot of outgoing vectors point to another general direction. It can also exhibit no clear directionality. In section 4 we describe how these evaluation results are interpreted and used to identify most promising areas of the solution space.

Figure 1 shows schematically a situation where plan B compared favorably to plan A and unfavorably to plans C and D. Plan E did not give a clear answer to comparisons with any other plan. This example uses two design parameters (plans represented in two dimensions) and just one heuristic comparison criterion (at most one evaluation vector is present for each pair of plans).

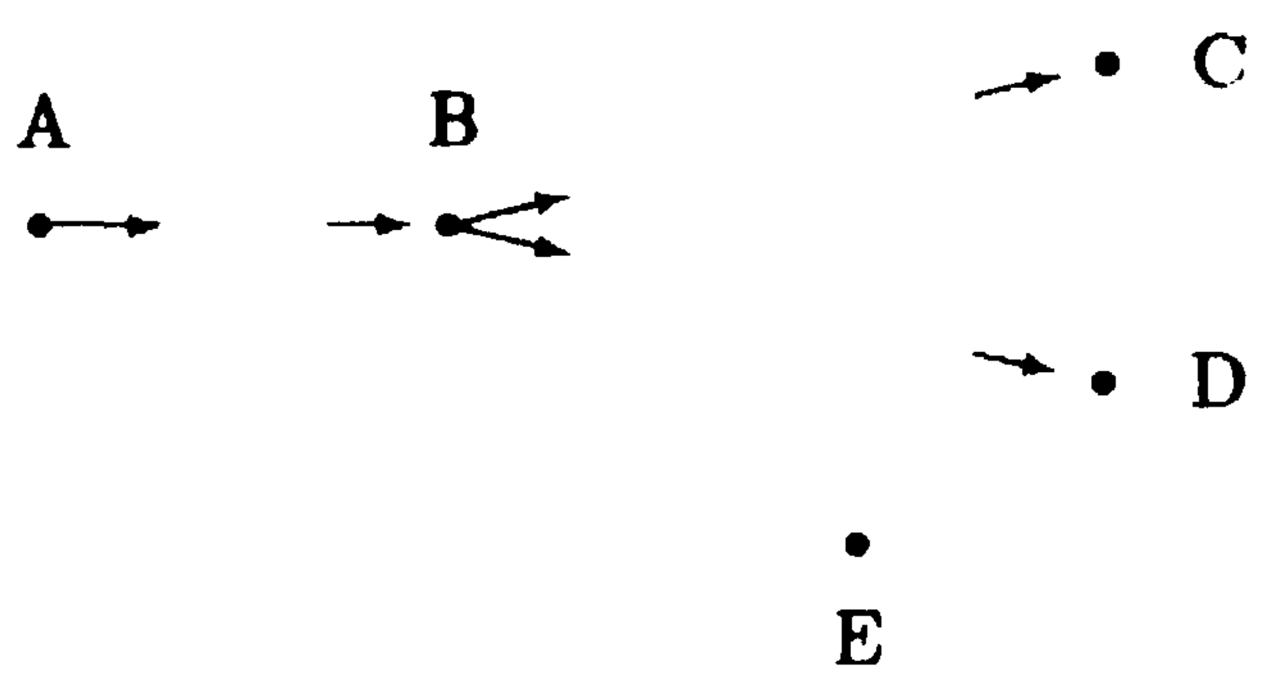

Figure 1: Dynamic evaluation results

Ill brief, in this section we described some ways of evaluating the inherently hard to evaluate radiation treatment plans. We showed that by giving up the scalar property of the evaluation results we can have very meaningful and intuitive results which are directly useful in the treatment plan optimization cycle (to generate plan "fixes"). We also introduced the notion of a *dynamic evaluation function* and showed how treatment plans could be compared in a meaningful way. This function gives only an indication of which candidate solutions are better but does not provide the absolute scores. While we know that it can in many cases be computed we have not yet shown how its outcome can be interpreted to control the search (see section 4 below).

3 Search space and the similarity metric

Our approach to the problem described here requires an explicit representation of the solution space. We need to see the potential therapy plans as points in this space and to have a "similarity metric" for deciding whether two plans are "similar" or not.

In principle it is not hard to define such a metric when dealing with continuously-varying parameters. However, our search space is somewhat complicated by the presence of some discrete parameters like the radiation type or wedge filters and blocks inserted in the beam path to modify the dose profile in some desired way. Also, dealing with plans with different number of radiation beams causes the solution space to have a varying number of dimensions. (Each beam has a number of characteristics like a particular direction, cross section size and shape, radiation type, energy, treatment time, etc.).

$$d(X,Y) \overset{\text{def}}{=} \max_{\substack{\text{over all}\\ \text{beams } B}} \max_{\substack{\text{over all}\\ \text{parameters } P\\ \text{of beam } B}} |X_{B_P} - Y_{B_P}| \times \text{sc\_fact}_P$$

The similarity metric, computed according to the above formula, is the maximum of the scaled parameter difference over all the parameters. This way, if two different plans have a number of parameters slightly different then they can be considered similar whereas if even one parameter is significantly different then they are not. While all parameters are scaled some also require additional special treatment. For example, beam angles must be adjusted modulo 360°. When one plan has more beams than another then existing beams are compared to non-existing beams. In such case all parameters in the non-existing beam are taken to be the same as in the existing beam, except beam intensity (monitor units) which is taken to be 0.0. This way, if one plan only differs from another one by a beam of small intensity it can still be found to be close to the other one.

One difficulty in this scheme is in deciding which are the "corresponding" beams in two plans being compared. Of course, comparing different beam pairs in two plans will give different results. We decided the proper way to treat this case is to compute the distance for all possible beam combinations and taking the minimum distance.

In summary, we needed an explicit representation of the solution space to be able to see groups of similar treatment plans. There is a measure of "distance" (a metric) between points in this space. This metric conveys the same notion of "similarity" as the candidate generator and does not have to correspond in any way to the similarity of the results obtained from each candidate. Although each problem class like this one needs to have specific definition of a search space and a metric, the approach we describe should apply in general.

4 The similarity analysis of the solution space

The elements of our radiation treatment planning system, described above and elsewhere [Paluszyriski, 1989] are:

* a library of PROTOTYPES for rapidly approaching a reasonably-looking solution in many common cases of cancer,

* a SIMULATION SYSTEM computing the raw dose distribution results,

* a STATIC EVALUATION FUNCTION which pinpoints trouble spots (constraint violations) in a plan,

* EXPERT IMPROVEMENT RULES (triggered by the constraint violation results) for selecting the next modification to achieve some desired effect in the existing plan,

* a DYNAMIC EVALUATION FUNCTION program for heuristically comparing two different treatment plans.

What is still needed is the method of interpreting the dynamic evaluation results. The outcome of this analysis is needed to select the next most promising treatment plan candidate to be optimized, or decide that no such promising candidate exists. We can then operate a SELECT-GENERATE-TEST cycle to efficiently construct better and better plans. While we have already described highly specialized components to perform the GENERATE and TEST steps the SELECT question needs elaboration here.

A simple scheme to answer this question is described below. We first interpret the dynamic evaluation results to classify all explored treatment plans as either *successful* or *unsuccessful* by simply counting the incoming and outgoing dynamic evaluation vectors (see section 2.2 above). The proposed modifications are represented as unexplored points in the solution space, alongside the explored and evaluated points. We then define two distances which we call: similar-plans range (A) and essentially-same-plans range *(6).* Then, for a given point X we only analyze the surrounding areas (actually n-cubes because of the similarity formula) of size A and 6 considering the previously explored points existing in those areas. The following 5 cases are recognized:

Case 1 Explored points exist within δ.
> Priority: MEDIUM

Plan X probably does not represent much improvement but is worth pursuing in the final fine tuning phase. On rare occasions a small change in a parameter value can result in a big improvement in dose.

Case 2 No explored points within *8.* Some successful explored points within A.
l> Priority: HIGH

This is a promising unexplored area.

Case 3 No explored points within δ. No successful explored points within A. Some unsuccessful explored points within A.
Priority: LOW

This is an unpromising area. Explore it only after all other possibilities failed.

Case 4 No explored points within δ. No successful or unsuccessful explored points within A (there can exist some explored points but they could not be clearly evaluated as successful or not successful).
t> Priority: MEDIUM-LOW

No useful information seems possible to extract from the existing plans. Do not waste time unless all other places are just the same.

Case 5 No explored points within 6 or A.
l> Priority: MEDIUM-HIGH

This is a new area which will be worth exploring as soon as we are done with the hot places.

The above procedure is actually only a simplified version of a more elaborate heuristic we are currently investigating. On the one hand, one can use a pattern classification procedure (such as *clustering,* see fDuda and Hart, 1973]) to guide the exploration of the solution space by the clouds, or clusters, instead of fixed-size ranges. Priorities would then be assigned to clusters and any new proposed plans within high-priority clusters would be explored first. On the other hand, even more importantly, we are trying to extend the above scheme to take more advantage of dynamic evaluation vectors. While currently we are interpreting them only as indicators of success or failure of plans we would also like to use the directionality of the vectors (see section 2.2 above) to further improve the efficiency of the search.

The process described here is a little like a numerical search procedure where new points are in turn generated in the solution space, evaluated, and some point selected for further generation. Unlike the numerical methods, however, the new points are generated by a specialized rule based expert system which recommends qualitatively the "right" fix for any problem. So the nature of the traversal in the solution space is different. Nevertheless the whole process represents a combination of a rule based approach, which gives a good representation of the planner's expertise, with numerical methods that reduce the amount of search in a way that is complementary to the symbolic knowledge.

5 Results and conclusions

We are in the process of implementing the whole system with a rule base for head and neck cancers and treatments by the radiation therapy machines available at the University of Washington Cancer Center. The radiation treatment simulation programs we utilize [Kalet and Jacky, 1982] are used routinely by human expert dosimetrists as an aid in the process of designing treatments for University Cancer Center patients. These programs and the numerical procedures for the plan evaluation are written in Pascal while the symbolic reasoning system is written in Common Lisp. The whole system [Paluszyriski, 1989] is meant to be eventually used in aiding the physicians in obtaining therapy plans in daily practice. Physicians would still be able to use the plans as they wish, but the program would save them from doing the most time-consuming task, namely, exploring all the available options.

While the final results will not be available until the full-scale system is coded and tested on the most difficult cases, we have tested all its elements as separate programs on a small sample of treatment plans for 5 patient cases. As expected, the production system, even under the most favorable conditions, generated some number of dead-end plan improvement candidates. These dead-ends were at first easily prevented from further exploration, as long as all clearly better options were open. When they were finally explored they blocked further search around them because either they were un-evaluable or evaluated to failures. More importantly however, there were many cases where the plan similarity played a major role. Whenever there was a promising plan A and an unsuccessful modification of it B, the process of exploring the close neighborhood of A would invariably lead to exploring a corresponding neighborhood of B because the same rule that generated B in most cases also applied to all plans around A. However, whenever the first plan in B's neighborhood was evaluated and determined to be a failure all the others were also automatically prevented from being evaluated. This behavior is very desirable and resulted in the greatest saving in search time.

Several questions have come up that warrant further exploration. The scheme we describe works best if a dynamic score can be generated that is complete and reliable; its performance gradually degrades when the results become more and more sparse. It would be interesting to obtain some quantitative measure of this degradation (eg. how many more unsuccessful plans are explored if the evaluation results are only available in 10% vs. 50% of cases).

The dynamic evaluation technique is an open ground for experimentation. As mentioned above, interpreting the directionality of success/failure vectors leads to more priority levels and thus better control of the search. But since the current evaluation function is based on simple heuristics there is a great potential for further refining it. Such a function would be invaluable in radiation oncology not only to speed up the treatment planning process but even to develop and study new approaches to radiation treatment. Developing new treatment approaches, such as 3-D treatments is mostly constrained by difficulties in evaluating such treatments by humans.

The practical implications of creating a therapy design system that performs well are clear, in that a much greater range of plans can be considered for each patient, in searching for an acceptable or optimal plan. Furthermore, this approach can easily be generalized to apply to other multi-dimensional, heterogeneous and ill-conditioned design problems and we can also expect to achieve good results.

Acknowledgements

It is a pleasure to acknowledge much illuminating consultation from Sharon Hummel and Dr. Mary Austin-Seymour concerning the intricacies of radiation treatment planning.

References

[Adeli and Balasubramanyam, 1988] H. Adeli and K. V. Balasubramanyam. A novel approach to expert systems for design of large structures. *AI Magazine,* 9(4):54-63, Winter 1988.

[Bleehen *et al,* 1983] N orman M. Bleehen, Eli Glatstein, and John L. Haybittle, editors. *Radiation Therapy Planning.* Marcel Dekker, Inc., New York, 1983.

[Brown and Chandrasekaran, 1986] David C. Brown and B. Chandrasekaran. Knowledge and control for a mechanical design expert system. *Computer,* 19(7):92-100, July 1986.

[Duda and Hart, 1973] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley, 1973.

[Kalet and Jacky, 1982] Ira J. Kalet and Jonathan P. Jacky. A research oriented treatment planning program system. *Computer Programs in Biomedicine,* 14(1):85-98, 1982.

[Kalet and Paluszyriski, 1985] Ira J. Kalet and Witold Paluszyriski. A production expert system for radiation therapy planning. In Allan H. Levy and Ben T. Williams, editors, *Proceedings of the Congress on Medical Informatics,* pages 315-319, San Francisco, 1985. American Association for Medical Systems and Informatics (AAMSI).

[Marcus *et* a/., 1988] Sandra Marcus, Jeffrey Stout, and John McDermott. VT: An expert elevator designer that uses knowledge-based backtracking. *AI Magazine,* 9(1):95-112, Spring 1988.

[Mittal and Araya, 1986] Sanjay Mittal and Agustin Araya. A knowledge-based framework for design. In *Proceedings of the National Conference on Artificial Intelligence,* pages 856-865, Philadelphia, Pennsylvania, August 1986.

[Mostow, 1985] Jack Mostow. Toward better models of the design process. *AI Magazine,* 6(l):44-57, Spring 1985.

[Paluszyriski and Kalet, 1987] Witold Paluszyriski and Ira J. Kalet. Applying expert system techniques to design problems. In *Proceedings of the Second Western Expert Systems Conference,* pages 169-176, Anaheim, California, 1987.

[Paluszyriski, 1989] Witold Paluszyri ski. *Designing radiation therapy for cancer, an approach to knowledge--based optimization.* PhD thesis, University of Washington, 1989. Forthcoming.

[Wolbarst *et al,* 1982] Anthony B. Wolbarst, Lee M. Chin, and Goran K. Svensson. Optimization of radiation therapy: Integral response of a model biological system. *International Journal Radiation Oncology Biology Physics,* 8(10): 1761-1769, 1982.