Ultra-fast fluence optimization for beam angle selection algorithms

M Bangert, P Ziegenhein, and U Oelfke
German Cancer Research Center - DKFZ
Im Neuenheimer Feld 280, 69120 Heidelberg, Germany
E-mail: m.bangert@dkfz.de

Abstract. Beam angle selection (BAS) including fluence optimization (FO) is among the most extensive computational tasks in radiotherapy. Precomputed dose influence data (DID) of all considered beam orientations (up to 100 GB for complex cases) has to be handled in the main memory and repeated FOs are required for different beam ensembles. In this paper, the authors describe concepts accelerating FO for BAS algorithms using off-the-shelf multiprocessor workstations. The FO runtime is not dominated by the arithmetic load of the CPUs but by the transportation of DID from the RAM to the CPUs. On multiprocessor workstations, however, the speed of data transportation from the main memory to the CPUs is non-uniform across the RAM; every CPU has a dedicated memory location (node) with minimum access time. We apply a thread node binding strategy to ensure that CPUs only access DID from their preferred node. Ideal load balancing for arbitrary beam ensembles is guaranteed by distributing the DID of every candidate beam equally to all nodes. Furthermore, we use a custom sorting scheme of the DID to minimize the overall data transportation. The framework is implemented on an AMD Opteron workstation. One FO iteration comprising dose, objective function, and gradient calculation takes between 0.010 s (9 beams, skull, 0.23 GB DID) and 0.070 s (9 beams, abdomen, 1.50 GB DID). Our overall FO time is < 1 s for small cases, larger cases take ~ 4 s. BAS runs including FOs for 1000 different beam ensembles take ~ 15–70 min, depending on the treatment site. This enables an efficient clinical evaluation of different BAS algorithms.

1. Introduction
Automated beam angle selection (BAS) incorporating fluence optimization (FO) requires repeated FOs in order to compare different beam ensembles. Relying on commercially available treatment planning software, which may require a couple of minutes per FO, however, would result in prohibitive computation times. In order to accelerate FO, BAS approaches including FO often compromise regarding the resolution in the dose deposition space [1] or regarding the number of evaluated beam ensembles [2]. Furthermore, the number of considered candidate beam orientations is often restricted [3] because precomputed dose influence data (DID) of all candidate beams has to be handled in higher memory.

We have recently investigated the structure of the combinatorial BAS problem [4] and the performance of different BAS algorithms for intracranial IMRT [5]. Therefore, we have developed an ultra-fast BAS framework operating on the same resolution like clinical planning systems, considering up to 1400 candidate beams, and evaluating up to 10 000 beam ensembles per patient. Here, we present a detailed description of the underlying FO algorithm and the hard- and software implementation of our BAS framework.
2. Material and methods

BAS strategies including FO usually comprise two independent modules, as depicted in figure 1. A BAS module controls which beam ensembles $B_i$ will be evaluated by the FO module in iteration $i$. In subsequent iterations $i + 1$, the BAS module suggests new beam ensembles based on the preceding objective function values $F(B_i)$. Choices for BAS modules are manifold and include simulated annealing [3], genetic algorithms [6], and iterative search strategies [7], among others. Usually, the runtime of the BAS module is orders of magnitude shorter than the runtime of the FO module. Hence, accelerating the entire BAS algorithm is only possible by accelerating the FO.

![Figure 1. Schematic software design of beam angle selection algorithms.](image)

2.1. Algorithmic implementation

We use a quadratic objective function $F$ [8] for FO

$$F = \sum_{i \in \text{Target}} p_i (D_i - D_i^{\text{pres}})^2 + \sum_{i \in \text{OAR/NT}} p_i \lfloor D_i - D_i^{\text{max}} \rfloor_+^2$$  \hspace{1cm} (1)

where $p_i$, $D_i$, and $D_i^{\text{pres}/\text{max}}$ denote the penalty, the dose, and the prescribed/maximum dose of voxel $i$. The positivity operator ensures that only violated constraints contribute to the objective function, i.e. $\lfloor x \rfloor_+ = x$ for $x > 0$ and $\lfloor x \rfloor_+ = 0$ otherwise. The dose in voxel $i$ is given by $d_i = \sum_j D_{ij}w_j$ with a DID matrix $D_{ij}$ specifying the dose contribution from bixel $j$ to voxel $i$. Minimizing equation 1 is a constrained convex optimization problem [9, Chapter 2.1.2.] as we require $w_j \geq 0$ for all beamlets $j$. Usually, such optimization problems are solved by sequential quadratic programming (SQP). However, we use a simple projected Quasi-Newton optimizer [10] as the improved algorithmic convergence of SQP is compromised by the additional arithmetic operations required during the solution of the quadratic subproblem. We apply an L-BFGS two-loop recursion algorithm to approximate the product of the inverse Hessian and the gradient [10]. An Armijo backtracking line search controls the step length in every iteration [10]. If a beam orientation $\beta_k$ is used for the first time during BAS, its beamlet weights are uniformly initialized $w_{jk} = 1$. If $\beta_k$ is used again during BAS as part of a different beam ensemble, its beamlet weights are initialized with the result of the last optimization. This accelerates the FO roughly by a factor of two. Even though the initial beamlet weights of $\beta_k$ have been obtained in combination with other beams, this does not affect the result of the FO due to the convexity of the FO problem.

2.2. Hard- and software implementation

The objective function and gradient calculation of equation 1 correspond to simple matrix vector products yet on large DID sets $D_{ij}$. Hence, FO is a memory bound problem [11]. The arithmetic load of the CPU does not limit the runtime of the algorithm but the transportation of the dose...
influence data $D_{ij}$ from the main memory to the CPU, which typically amounts to 100 MB - 1 GB for a single beam ensemble. On state-of-the-art multiprocessor workstations the speed of data transportation from the main memory to the CPUs is non-uniform across the RAM; every CPU has a dedicated memory location (node) with minimum access time. We apply a thread node binding strategy to ensure that CPUs only access DID from their preferred node. The DID of every candidate beam is distributed equally to all nodes. Hence, we guarantee an ideal load balancing for every potential beam ensemble that might be evaluated during BAS. Furthermore we sort the DID according to the access pattern during the objective function and gradient calculation in order to minimize the overall data transportation of base data between RAM and CPU [12]. Efficient cache memory filling is ensured by always processing chunks of 64 bytes, which corresponds to the standard cache line size, at the same time. We use the OpenMP standard for parallelization in C++. The BAS framework is implemented on an AMD Opteron workstation with 4 CPUs with 12 cores each and 128 GB main memory. Besides a small overhead required for the optimization, the entire main memory is available to store precomputed DID.

2.3. Beam angle selection study

We measure the runtimes of BAS runs comprising 10,000 FOs for three prostate, three pancreas, and three intracranial cases. We compare treatment plans applying nine non coplanar beam orientations that are found with a genetic algorithm [5] and treatment plans applying nine equi-spaced coplanar beams; a more detailed analysis of the clinical implications of our BAS framework for intracranial IMRT can be found in [5].

3. Results

Figures 2 (a)-(c) summarize the results of the performance measurements of BAS runs for nine different patient cases with 10,000 FOs each. As depicted in figure 2(a), we require 10-70 ms per FO iteration depending on the problem size, i.e., the amount of DID for a given beam ensemble. One iteration comprises one gradient calculation and on average 1.09 line searches. As depicted in figure 2(b), the number of FO iterations is usually around 50 if the FO is terminated after three successive iterations with an objective function decrease of less than 0.1%. The number of iterations does not depend on the amount of DID processed during the FO. The runtime of a single FO is 0.5-4.0 s depending on the problem size, as depicted in figure 2(c).

Figure 2(a) shows that the minimum runtime linearly depends on the amount of DID. The fluctuations can be explained by the individual locality of the DID in the RAM, by the size of the DID, and in figure 2(c) also by the number of iterations.

Figures 2 (d)-(f) show representative DVHs for one prostate, pancreas, and intracranial case. We observe modest reductions of the dose in organs at risk (OARs) for the prostate and pancreas cases at similar target coverage. For the intracranial cases BAS may enable substantial dose savings in OARs.

4. Discussion and conclusion

We have presented an ultra-fast FO implementation for BAS that requires about 15-70 min to compare 1,000 treatment plans. This enables an efficient evaluation of different BAS strategies for clinical application [5, 4].

The fast BAS runtimes originate from a CPU-based FO engine which needs less than 1 s for small cases. Men et al. report runtimes of 2.79 s on a GPU for FO problems of comparable size [13].

The developed ultra-fast FO engine may be useful for manifold applications beyond BAS as it is generally suited to accelerate inverse planning. Naturally, a speedup is especially desirable for time consuming treatment planning tasks involving large DID sets > 1 GB such as modulated arc therapy and particle therapy. For small data sets ~ 0.1 GB it is possible to enable a more
Figure 2. (a) Runtime of one FO iteration versus the amount of processed DID. (b) Number of required FO iterations. (c) Overall FO runtime versus the amount of processed DID. The nine different colors in figures (a) and (c) correspond to different patient cases. The point clouds comprise 10 000 dots each; a single dot corresponds to a single FO. DVHs for (d) a prostate case, (e) a pancreas case, and (f) an intracranial case. Solid lines correspond to a treatment plan applying nine equi-spaced coplanar beams and dotted lines correspond to nine non coplanar optimized beams.

interactive treatment planning experience - and maybe ultimately realtime FO. For adaptive radiation therapy, an acceleration of inverse planning is an indispensable prerequisite before clinical application because a treatment plan of the day has to be generated based on up-to-date imaging data while the patient is on the couch.

Even though our implementation relies on the standard quadratic objective function [8], the acceleration concepts are generally applicable to treatment planning approaches that use a matrix vector product for the dose calculation. This includes alternative objective functions, multicriteria optimization, and direct aperture optimization, among others.

In this study, we have used an AMD workstation with 128 GB main memory in order to cope with the DID of all candidate beams. However, our implementation can also be used to accelerate FO on conventional personal computers [11].

An efficient application of GPUs in the context of BAS is currently not possible. Due to the limited memory available on GPUs, the DID of all candidate beams cannot be loaded with a single GPU. An efficient application of GPU clusters is infeasible due to synchronization and load balancing issues. Furthermore GPU clusters are very expensive; the AMD workstation we applied for this study is available at US$ 4500.

References

[1] Engel K and Tabbert E 2005 Optimization and Engineering 6 393–419
[2] Rowbottom C G, Khoo V S and Webb S 2001 Medical Physics 28 1696
[3] Stein J, Mohan R, Wang X H, Bortfeld T, Wu Q, Preiser K, Ling C C and Schlegel W 1997 Medical Physics 24 149
[4] Bangert M, Ziegenhein P and Oelfke U 2012 Physics in Medicine and Biology 57 6707
[5] Bangert M, Ziegenhein P and Oelfke U 2013 Medical physics 40 011716
[6] Hou Q, Wang J, Chen Y and Galvin J M 2003 Medical Physics 30
[7] Woudstra E, Heijmen B J and Storchi P R 2005 Radiotherapy and Oncology 77
[8] Oelfke U and Bortfeld T 2001 Medical Dosimetry 26 113–124
[9] Bangert M 2011 New concepts for beam angle selection in IMRT treatment planning: From heuristics to combinatorial optimization Ph.D. thesis Universitätsbibliothek Heidelberg
[10] Kelley C T 1987 Iterative methods for optimization vol 18 (Society for Industrial and Applied Mathematics)
[11] Ziegenhein P, Kamerling C, Bangert M, Kunkel J and Oelfke U 2013 Physics in medicine and biology 58 3705–15
[12] Ziegenhein P, Wilkens J, Nill S, Ludwig T and Oelfke U 2008 Physics in medicine and biology 53
[13] Men C, Gu X, Choi D, Majumdar A, Zheng Z, Mueller K and Jiang S B 2009 Physics in medicine and biology 54 6565