# Fixed-Parameter Tractability and Improved Approximations for Segment Minimization*

**Technical Report CS-2009-03**

**January 20, 2009**

Therese Biedl[1], Stephane Durocher[1], Holger H. Hoos[2], Shuang Luan[3], Jared Saia[3], and Maxwell Young[1]

[1] David R. Cheriton School of Computer Science, University of Waterloo, ON, Canada
{biedl@uwaterloo.ca,sdurocher@uwaterloo.ca,m22young}@uwaterloo.ca
[2] Department of Computer Science, University of British Columbia, BC, Canada
hoos@cs.ubc.ca
[3] Department of Computer Science, University of New Mexico, NM, USA
{sluan, saia}@cs.unm.edu

**Abstract.** The segment minimization problem consists of finding the smallest set of integer matrices that sum to a given intensity matrix, such that each summand has only one non-zero value, and the non-zeroes in each row are consecutive. This has direct applications in intensity-modulated radiation therapy, an effective form of cancer treatment.

We show here that for a single row, this problem is fixed-parameter tractable in the largest value of the intensity matrix. We use this to develop approximation algorithms for the full problem. One of these improves the approximation factor from the previous best of $\log_2 h + 1$ to $3/2 \cdot (\log_3 h + 1)$, where $h$ is the largest entry in the intensity matrix; another improves the approximation factor from $2 \cdot (\log D + 1)$ to $24/13 \cdot (\log D + 1)$, where $D$ is the largest difference between consecutive elements of a row of the intensity matrix.

Experimentation with these algorithms show that they outperform other approximation algorithms on 75% of the 172 test cases we considered, which include both real world and synthetic data.

## 1 Introduction

Intensity-modulated radiation therapy (IMRT) is an effective form of cancer treatment, in which the region to be treated is discretized into a grid, and a treatment plan specifies the amount of radiation to be delivered to the area of body surface corresponding to each grid cell. A device called a multileaf collimator (MLC) is used to administer the treatment plan in a series of steps. In each step, two banks of metal leaves in the MLC are positioned to cover certain portions of the body surface, while leaving others exposed, and the latter are then subjected to a specific amount of radiation.

A treatment plan can be represented as an $m \times n$ *intensity matrix* $T$ of non-negative integer values, whose entries represent the amount of radiation to be delivered to the

* Research partially supported by NSERC. We also thank Baruch Schieber for his helpful correspondence.

corresponding grid cells. The leaves of the MLC can be seen as partially covering rows of $T$; for each row $i$ of $T$ there are two leaves, one of which may slide inwards from the left to cover the elements in columns $1..l$ of that row, while the other may slide inwards from the right to cover the elements in columns $r..n$. After each step of the treatment, the amount of radiation applied in that step (this can differ per step) is subtracted from each entry of $T$ that has not been covered. The treatment is completed when all entries of $T$ have reached $0$.

Setting leaf positions in each step of the treatment plan requires time. Minimizing the number of steps reduces treatment time and can result in increased patient throughput, reduced machine wear and tear, and overall reduced cost of the procedure. Minimizing the number of steps for a given treatment plan is the objective of this paper.

Formally, a *segment* is a matrix $S$ such that non-zeroes in each row of $S$ are consecutive, and all non-zero entries of $S$ are the same integer, which we call the *segment-value*. A *segmentation* of $T$ is a set of segment matrices that sum up to $T$, and we call the cardinality of such a set the *size* of that segmentation. The *segmentation problem* is, given an intensity matrix $T$, to find a minimum-size segmentation of $T$. We will often consider the special case of a matrix $T$ with one row, which we call the *single-row segmentation problem* as opposed to the *full-matrix segmentation problem.* For ease of notation, we assume that $T$ *begins and ends with an all-0 column*; this does not affect the size of an optimal segmentation and simplifies the definition of $D$ given in [14].

The segmentation problem is known to be NP-complete in the strong sense, even for a single row [6, 1, 2], as well as APX-complete [3, 4]. Numerous heuristics are known [13, 8, 11, 12]. Bansal *et al.* [3, 4] provide a $24/13$-approximation algorithm for the single-row problem and give some better approximations for more constrained versions. Work by Collins *et al.* [7] shows that the single *column* version of the problem is NP-complete and provides some non-trivial lower bounds given certain constraints. Recent work by Luan *et al.* [14] gives two approximation algorithms for the full $m \times n$ segmentation problem; however, they do not consider the performance of their algorithms in practice.

**Our Contributions**

Luan et al. [14] used two insights to obtain approximation algorithms. First, the segmentation problem is easy for 0/1-matrices. Second, segmentations for the single-row problem with small segment-values can be used to obtain good segmentations for the full-matrix problem. They exploited both and gave two approximation algorithms with approximation factors of (roughly) $\log h$ and $2 \log D$ where $h$ is the largest value in $T$, and $D$ is the largest difference between consecutive elements in a row of $T$.[4]

We use the same ideas, but add further insights. First, we show that the single-row segmentation problem is *fixed-parameter tractable* in the largest value $h$ (i.e., the runtime is $O(f(h)p(n))$ for some function $f(.)$ and some polynomial $p(.)$). Hence, the single-row problem is easy to solve if $h$ is small. Unfortunately, this does not immediately imply that the full-matrix problem is easy to solve if $h$ is small, but we can solve it optimally in polynomial time for $h = 2$. With some further insight, we can show that such solutions can be combined to give an approximation algorithm for the full-matrix

[4] Throughout this work, we use $\log(x)$ to mean $\lfloor \log x \rfloor$. Unless specified otherwise, we assume the logarithm base $2$.

segmentation problem with approximation factor (roughly) $\frac{3}{2} \cdot \log_3(h)$, which is smaller than $\log h$.

We also provide another approximation algorithm with factor (roughly) $\alpha \log D$, where $\alpha$ is the best approximation factor for the single-row problem. The current best known $\alpha$ is $\alpha = 24/13$ [3, 4]; any improved approximation result for the single-row problem would lead directly to an improved approximation result for the full problem. This second approximation algorithm expands on the second approximation algorithm by Luan et al.; they used one specific 2-approximation algorithm for the single-row problem, whereas we show that in fact any $\alpha$-approximation algorithm can be used.

Finally, we give an empirical evaluation of known approximation algorithms, using both synthetic and real-world clinical data. To the best of our knowledge, this is the first such evaluation of these approximation algorithms to appear in the literature. Our experiments demonstrate that the constant factor improvements made by our algorithms yield significant performance gains in practice. Therefore, in both the $O(\log h)$ and $O(\log D)$ scenarios, our new algorithms improve on previous approximation algorithms *theoretically and experimentally*.

## 2 FPT algorithms for single-row segmentation

In this section, we prove that the single-row segmentation problem is fixed parameter tractable (FPT) in $h$, the largest value in the intensity matrix $T$. Note that $T$ is has a single-row, hence it is a string $T[1..n]$. We call a segmentation of $T[1..n]$ *compact* if any two segments in it *begin* (i.e., have their first non-zero entry) at a different index, and *end* (i.e., have their last non-zero entry) at a different index. The following observation is straightforward; we give a proof in the appendix.

**Lemma 1.** *For any segmentation $\mathcal{S}$ of a single row, there exists a compact segmentation $\mathcal{S}'$ with $|\mathcal{S}'| \leq |\mathcal{S}|$.*

Our algorithm uses a dynamic programming approach that computes an optimal segmentation of any prefix $T[1..i]$ of $T$. We say that a segmentation of $T[1..i]$ is *almost-compact* if any two segments in it either begin at different indices or both begin at index 1, and they either end at different indices or both end at index $i$. We will only compute almost-compact segmentations; this is sufficient by Lemma 1. We compute the segmentation conditional on the values of the last segments in it.

Let $\mathcal{S}$ be a segmentation of string $T[1..i]$; each $S \in \mathcal{S}$ is hence a string $S[1..i]$. Define the *signature* of $\mathcal{S}$ to be the multi-set obtained by taking the last integer $S[i]$ of each segment $S \in \mathcal{S}$ and deleting all 0s. Note that the signature of a segmentation of $T[1..i]$ is a *partition* of $T[i]$, i.e., a multi-set of positive integers that sum to $T[i]$. For any partition $\phi_i$, use $||\phi_i||$ to denote its size, i.e., the number of elements, counting multiple elements repeatedly.

Now define a function $f$ as follows: given an integer $i$ and a partition $\phi_i$ of $T[i]$, set $f(i, \phi_i)$ to be the minimum number of segments in an almost-compact segmentation of $T[1..i]$ for which the signature is $\phi_i$. One can easily see that $f(1, \phi_1) = ||\phi_1||$. We will show that $f(i, \phi_i)$ can be computed recursively. Given a partition $\phi_i$ of $T[i]$, let $\Phi_{i-1}(\phi_i)$ be the set of those partitions of $T[i-1]$ that can be obtained from $\phi_i$ by deleting at most one element, and then adding at most one element.

**Lemma 2.** *For* $i > 1$, $f(i, \phi_i) = \min_{\phi_{i-1} \in \Phi_{i-1}(\phi_i)} \{f(i-1, \phi_{i-1}) + ||\phi_i - \phi_{i-1}||\}$

*Proof.* We only prove "$\geq$" here; the other inequality is proved similarly (see appendix.) Consider an almost-compact segmentation $\mathcal{S}_i$ of $T[1..i]$ that achieves the left-hand side, i.e., its signature is $\phi_i$ and $|\mathcal{S}_i| = f(i, \phi_i)$. We have four kinds of segments in $\mathcal{S}_i$: (1) Those that end at index $i-2$ or earlier, (2) those that end at $i-1$ (there can be at most one, since $\mathcal{S}_i$ is almost-compact), (3) those that end at $i$ and start at $i-1$ or earlier, and (4) those that end at $i$ and begin at $i$ (there can be at most one).

Let $\mathcal{S}_{i-1}$ be the segmentation of $T[1...i-1]$ obtained from $\mathcal{S}_i$ by taking all segments of type 1–3, and deleting the last integer (at index $i$). Note that $\mathcal{S}_{i-1}$ is also almost-compact. The signature $\phi_{i-1}$ of $\mathcal{S}_{i-1}$ is the same as $\phi_i$, except all values of segments of type (4) are removed and all values of segments of type (2) are added. This shows that $\phi_{i-1}$ is in $\Phi_{i-1}(\phi_i)$.

If both a segment of type (4) and a segment of type (2) exist in $\mathcal{S}_i$, then they necessarily have different non-zero value (otherwise they could be combined, contradicting the minimality of $\mathcal{S}_i$). Hence $||\phi_i - \phi_{i-1}||$ is exactly the number of segments of type (4). So $|\mathcal{S}_{i-1}| = |\mathcal{S}_i| - ||\phi_i - \phi_{i-1}||$, which proves "$\geq$". □

By evaluating function $f$ with standard dynamic programming approaches, we can show the following:

**Theorem 1.** *The single-row segmentation problem can be solved in* $O(h^{1.5} \cdot p(h) \cdot n)$ *time and* $O(h \cdot p(h))$ *space if all values in the intensity matrix are at most* $h$*, where* $p(h)$ *is the number of partitions of integer* $h$.

*Proof.* (Sketch) For each $i$, there are at most $p(h)$ partitions of $T[i] \leq h$; computing and storing them can be done in $O(h \cdot p(h))$ time and space. Any partition of $T[i] \leq h$ has at most $\sqrt{h}$ many distinct integers; hence $|\Phi_{i-1}(\phi_i)| \leq \sqrt{h}$. So we can compute $f(i, \phi_i)$ in $O(h^{1.5} \cdot p(h))$ time if $f(i-1, .)$ is known. Doing this for all $i$, we can compute $f(n, \phi_n)$ for all partitions $\phi_n$ of $T[n]$ in time $O(h^{1.5} \cdot p(h) \cdot n)$, and the optimal segmentation-size is found by taking the minimum. □

It is known that $p(h) \leq e^{\pi \cdot \sqrt{\frac{2 \cdot h}{3}}}$ [10], so this algorithm is polynomial as long as $h \in O(\log^2 n)$. In the present form it only returns the size of the smallest segmentation, but standard dynamic programming techniques can be used to retrieve the segmentation in the same running time with an $O(\log n)$ space overhead.

### 2.1 The special case of $h = 2$

For $h = 2$, we can find the optimal solution with the above dynamic programming algorithm. However, we can do more: we can control how many segments have value 2 (we call these *2-segments*) and how many have value 1 (we call these *1-segments*.) These results will be needed later when we combine solutions in each row to a solution of the whole matrix.

We can use regular expressions to describe subsequences of $T$, e.g., $2^+$ stands for 'a subsequence of only 2s, containing at least one 2'. Let a *step* be a subsequence of $T$ of the form $02^+1$ or $12^+0$ and a *tower* be a subsequence of $T$ of the form $02^+0$. See also Fig. 1. A *marker* is an index $i$ for which $T[i-1] \neq T[i]$ ([14]; this was called *tick* in [3].) We use $s$, $t$ and $\rho$ for the number of steps, towers, and markers, respectively.

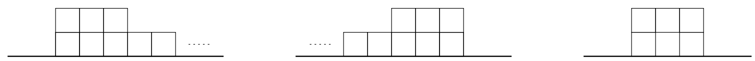


**Fig. 1.** Two kinds of steps, and a tower.

**Lemma 3.** *Define $g(d)$ as follows:*

$$g(d) = \begin{cases} \frac{1}{2}\cdot(\rho+s) - d & \text{if } t \le d \le t+s \\ \frac{1}{2}\cdot(\rho+s) - (t+s) & \text{if } t+s < d \\ \frac{1}{2}\cdot(\rho+s) + t - 2d & \text{if } d < t \end{cases}$$

*Then for any $d \geq 0$, any segmentation with at most $d$ 2-segments has at least $g(d)$ 1-segments. Moreover a segmentation that has at most $d$ 2-segments and exactly $g(d)$ 1-segments can be found in $O(n)$ time.*

*Proof.* Any occurrence of a 2 is at a tower, at a step, or at a substring of the form $12^+1$. Applying dynamic programming, one verifies the following:

- Towers must use a 2-segment in any optimal solution. For each tower that does not use a 2-segment, the size of the solution increases by 1.
- Steps may use a 2-segment in an optimal solution, but they can also use a 1-segment instead without affecting the size of the solution.
- Substrings of the form $12^+1$ cannot use a 2-segment in an optimal solution.

Hence, if we are allowed a 2-segment for each tower (i.e., $d \geq t$), then the best segmentation will have the optimum size $OPT$ (one can easily show that $OPT = (\rho+s)/2$ by evaluating all possible cases with dynamic programming.) At most $t+s$ 2-segments can be used in an optimal segmentation. So the number of 1-segments is $OPT - d$ if $d \leq t+s$ and $OPT - (t+s)$ if $d > t+s$.

Now assume that $d < t$. In this case, we are not even allowed a 2-segment for each tower, and $t - d$ of them must use two 1-segments instead. So the best solution uses $OPT + (t-d)$ segments, of which $d$ are 2-segments and the rest are 1-segments. To find such a segmentation, use a 2-segment for $\min\{d,t\}$ towers, then for $\min\{d-t, s\}$ steps if $d \geq t$, and use 1-segments for everything else. □

The second result is that we can always find a segmentation whose number of $v$-segments ($v = 1, 2$) is bounded in terms of the number of markers alone. This segmentation is not necessarily optimal, but knowing these bounds will allow us to show approximation bounds later.

**Lemma 4.** *There exists a segmentation such that the number of 1-segments is at most $\frac{1}{2}\cdot\rho$, and the number of 2-segments is at most $\frac{1}{4}\cdot\rho+\frac{1}{2}$.*

*Proof.* (Sketch) Use a 2-segment for half of the towers (rounded up) and half of the steps (rounded down), and use 1-segments for everything else. The proof then follows by counting markers carefully. Details are given in the appendix. □

# 3 Improved Approximations Using A Larger Base

The first approximation given by Luan et al. [14] worked as follows. Split the given intensity matrix $T$ into matrices $P_1, \ldots, P_k$ such that $T = \sum_{\ell=1}^{k}(2^\ell \cdot P_\ell)$ (by taking the bits of the base-2 representation of entries of $T$). A segmentation for $T$ can then be obtained by taking segmentations of each $P_\ell$, multiplying their values by $2^\ell$, and taking their union. Since each $P_\ell$ is a 0/1-matrix, an optimal segmentation of it can be found easily, and an approximation bound of $\log h + 1$ holds.

We use exactly the same approach, but change the base, writing $T = \sum_{\ell=1}^{k}(b^\ell \cdot P_i)$ for some integer $b \geq 3$. (This can be done with any $b$, but we obtained good approximation bounds only for $b = 3, 4$.) This raises two questions: (1) How can we solve the segmentation problem in a matrix that has values in $\{0, 1, \ldots, b-1\}$? (2) Is the resulting segmentation a good approximation of the optimal segmentation? Neither of these questions is anywhere near as simple as it was for $b = 2$.

## 3.1 Splitting $P$ into rows and combining

We now address the first question, i.e., how to find a good segmentation of a matrix $P$ with values in $\{0, \ldots, b-1\}$. A simple heuristic consists of splitting $P$ into its rows, solving the segmentation problem for each row, and combining those segmentations into one segmentation $\mathcal{S}$. Since each row also has values in $\{0, \ldots, b-1\}$, we can find the optimum segmentation of each row in polynomial time (as long as $b$ is a constant.) To combine the rows, one can use a greedy approach. Check for each value $v \in \{0, \ldots, b-1\}$ whether any segment in any row has this value. If there is one, then remove a segment of value $v$ from each row that has one. Combine all these segments into one segment-matrix (also with value $v$), and add it to $\mathcal{S}$. We refer to this algorithm as GREEDYROWPACKING.

Using an optimal segmentation of each row seems a natural idea, but somewhat counter-intuitively, it sometimes isn't the best we can do. Consider an example where the optimal segmentation of row 1 uses 10 1-segments and no 2-segments, and the optimal segmentation of row 2 uses no 1-segments and 5 2-segments. GREEDYROWPACKING would then use 15 segments for $P$. But if instead we had used a different segmentation of row 2, which splits each 2-segment into two 1-segments, then both rows used 10 1-segments, and GREEDYROWPACKING would use 10 segments for $P$. So it is sometimes advantageous to use segmentations that are not optimal. This will be exploited for $b = 3$ below.

## 3.2 The case $b = 3$

If $b = 3$, then $P$ is a 0/1/2-matrix, i.e., all entries in $P$ are 0,1 or 2. For each row of $P$, we know not only how to compute the optimal segmentation, but for any $d$ we can compute the best segmentation that has at most $d$ 2-segments (Lemma 3). Let $g_i(.)$ be the function $g(.)$ as in Lemma 3 for row $i$. Thus we know that any segmentation of row $i$ with at most $d$ 2-segments has at least $g_i(d)$ 1-segments, and a segmentation with at most $d$ 2-segments and exactly $g_i(d)$ 1-segments can be computed in $O(n)$ time.

**Lemma 5.** *Any optimal segmentation $\mathcal{S}^*$ of $P$ satisfies* $|\mathcal{S}^*| = \min_d\{d + \max_i\{g_i(d)\}\}$.

*Proof.* We first prove '$\leq$'. Let $d^*$ be the value that achieves the minimum. For each row $i$, find a segmentation with at most $d^*$ 2-segments and $g_i(d^*)$ 1-segments. Apply GREEDYROWPACKING to obtain a segmentation $\mathcal{S}$ of $P$. The number of 1-segments in $\mathcal{S}$ is $\max_i g_i(d^*)$, and the number of 2-segments is at most $d^*$, so $|\mathcal{S}| \leq \min_d\{d + \max_i\{g_i(d)\}\}$, and $|\mathcal{S}^*|$ can only be smaller.

For the other direction, let $\mathcal{S}^*$ be an optimal segmentation of $P$, and let $d^*$ be the number of 2-segments in it. For each $i$, the induced segmentation of row $i$ hence has at most $d^*$ 2-segments and by Lemma 3 at least $g_i(d^*)$ 1-segments. The row $i$ that maximizes $g_i(d^*)$ has at least $\max_i\{g_i(d)\}$ 1-segments, so $\mathcal{S}^*$ has at least $\max_i\{g_i(d)\}$ 1-segments. Since $\mathcal{S}^*$ has $d*$ 2-segments, this proves the claim. □

We can hence find the optimal segmentation of $P$ as follows. Compute function $g_i(.)$ for each row, then compute function $\max_i\{g_i(.)\}$, and then find the value $d^*$ that minimizes $d + \max_i\{g_i(d)\}$. This can all be done in $O(m \cdot n)$ time, since the functions $g_i$ are very simple. Compute for each row the best segmentation with at most $d^*$ 2-segments, and combine these segmentations with GREEDYROWPACKING; by Lemma 5 this gives the optimal segmentation for $P$.

**Theorem 2.** *The minimal segmentation of an intensity matrix with values in $\{0, 1, 2\}$ can be found in $O(m \cdot n)$ time.*

### 3.3 Combining segmentations of matrices

Now we address the second question posed earlier. Assume $T = \sum_{i=1}^{k}(b^\ell \cdot P_\ell)$ for matrices $P_1, \ldots, P_k$, where $k = \log_b h + 1$. Assume further that we have *$\alpha$-approximate segmentations* for each $P_\ell$, i.e., for each $\ell$ we have a segmentation $\mathcal{S}_\ell$ of $P_\ell$ that is within a factor $\alpha$ of the optimum, for some $\alpha \geq 1$. We *combine* these segmentations as follows: For each segment $S$ of $\mathcal{S}_\ell$, add $b^\ell \cdot S$ to $\mathcal{S}$. One easily verifies that $\mathcal{S}$ is a segmentation of $T$. For $b = 3$, we can show that this is a good approximation.

**Lemma 6.** *Assume each $P_\ell$ is a 0/1/2-matrix. Combining optimal segmentations $\mathcal{S}_1^*$, $\ldots, \mathcal{S}_k^*$ for matrices $P_1, \ldots, P_k$ gives a segmentation $\mathcal{S}$ for $T$ of size at most $\frac{3}{2} \cdot k \cdot OPT + \frac{1}{2} \cdot k$, where $OPT$ is the size of a minimal segmentation of $T$.*

*Proof.* Rather than arguing this directly, we argue via another segmentation of each $P_\ell$ which has some desirable properties. Let $\rho_\ell^i$ be the number of markers of row $i$ of matrix $P_\ell$. Recall that each row $i$ of $P_\ell$ has a segmentation $\mathcal{S}_\ell^i$ for which the number of 1-segments is at most $\frac{1}{2} \cdot \rho_\ell^i$ and the number of 2-segments is at most $\frac{1}{4} \cdot \rho_\ell^i + \frac{1}{2}$ (Lemma 4.) Let $\rho_\ell = \max_i \rho_\ell^i$ be the maximum number of markers within any row of $P_\ell$. Combining the segmentations $\mathcal{S}_\ell^i$ of the rows of $P_\ell$ with algorithm GREEDYPACKING gives a segmentation $\mathcal{S}_\ell$ of $P_\ell$ for which the number of 1-segments is at most $\frac{1}{2} \cdot \rho_\ell$ and the number of 2-segments is at most $\frac{1}{4} \cdot \rho_\ell + \frac{1}{2}$. The optimal segmentation $S_\ell^*$ of $P_\ell$ can only be smaller, so

$$|\mathcal{S}_\ell^*| \leq |\mathcal{S}_\ell| \leq \frac{3}{4} \cdot \rho_\ell + \frac{1}{2}.$$

Consider the optimal segmentation $\mathcal{S}^*$ of $T$. Let $i$ be the row of $T$ which has the maximal number $\rho$ of markers. Every segment in $\mathcal{S}^*$ can remove at most two markers in row

$i$, which proves $2|\mathcal{S}^*| \geq \rho$. Matrix $P_\ell$ can have a marker only if matrix $T$ has a marker in the same location, so $\rho_\ell \leq \rho \leq 2|\mathcal{S}^*|$ [14]. Putting it all together, we have

$$|\mathcal{S}| = \sum_{\ell=1}^{k} |\mathcal{S}^*_\ell| \leq \sum_{\ell=1}^{k} (\frac{3}{4} \cdot \rho_\ell + \frac{1}{2}) \leq \sum_{\ell=1}^{k} (\frac{3}{4} 2|\mathcal{S}^*| + \frac{1}{2})$$

which proves the result. □

The above result showed the approximation bound already for the segmentation obtained by packing the segmentations of the rows of Lemma 4 into matrices. We know that these segmentations aren't optimal if there are many towers, so using the optimal segmentation of each $P_\ell$ should given even better bounds in practice. We conclude by restating the result as a theorem.

**Theorem 3.** *There exists a polynomial-time algorithm that for any intensity matrix $T$ with maximum value $h$ finds a segmentation $\mathcal{S}$ of $T$ size at most $\frac{3}{2} \cdot (\log_3 h + 1) \cdot OPT + \frac{1}{2} \cdot (\log_3 h + 1)$, where $OPT$ is the size of a minimal segmentation of $T$.*

For large $OPT$ and $h$ values, the new approximation factor approaches $\frac{3}{2} \cdot (\log_3 h + 1)$; therefore, the ratio between this approximation and the $(\log h + 1)$-approximation of [14] approaches $\frac{3}{2 \log 3} \approx 0.946$. Hence, for sufficiently large $OPT$ and $h$, the new algorithm is superior.

### 3.4 Higher values of the base

One could consider a similar approach using larger bases, and in particular $b = 4$. Two complications arise. First, we do not know how to compute the optimum segmentation of a matrix with values in $\{0, 1, 2, 3\}$, unless it is a single row. For $b = 3$, this was done in Lemma 3, which expressed the number of needed 1-segments in terms of the number of allowed 2-segments. It is not clear whether this lemma can be generalized to some $(b - 1)$-dimensional function for larger $b$.

It is also not straightforward that even an optimal solution for each $P_\ell$ would yields an approximate solution for $T$. This was argued for $b = 2$ and $b = 3$ using markers. With an extensive case analysis, we can generalize Lemma 4 to $b = 4$ as well (the number of 3-segments is at most $\rho/6$), which gives an $\frac{11}{6} \cdot (\log_4(h) + 1)$-approximate segmentation. Preliminary experimental results indicated that using base $b = 4$ is no better than using base $b = 3$ in practice, and we did not pursue this approach further.

## 4 Approximation by modifying row-segmentations

Our previous approximation algorithm can be summarized as follows: split the intensity matrix by bits, split each resulting matrix into rows, segment each row and then put the segments together. The second approximation algorithm by Luan et al. [14] uses another approach that is in some sense reverse: split the intensity matrix into rows, segment each row, split each resulting segment into multiple segments by bits, and then put the segments together. The quality of this second approximation depends on two factors: the approximation guarantee and the largest value used by a segment in any of the row-segmentations. Without formally stating it in these terms, Luan et al. proved the following result:

**Lemma 7.** *[14] Assume that for any single-row problem we can find an $\alpha$-approximate solution where all segments have value at most $M$. Then we can compute in polynomial time an $\alpha(\log M + 1)$-approximate segmentation of $T$.*

Luan et al. used this by showing that any single-row problem has a 2-approximate solution where any segment has value at most the maximum difference $D$ between consecutive elements in a row.

We can slightly improve on this with two insights. First, *any* segmentation can be converted into a segmentation with values at most $D$, without adding any new segments. Secondly, values $\alpha < 2$ can be found, both based on existing results and because of our first approximation algorithm.

**Lemma 8.** *Let $\mathcal{S}$ be any segmentation of a single-row intensity matrix $T$. Let $D$ be the maximum difference between consecutive elements in $T$. Then there exists a segmentation $\mathcal{S}'$ with $|\mathcal{S}'| \leq |\mathcal{S}|$ for which all segments have value at most $D$.*

*Proof.* Modify $\mathcal{S}$ as done in [2] such that no two segments meet, i.e., if some segment ends at index $i$, then no segment starts at $i+1$. Any segment $S$ must have value $v \leq D$, for if $S$ ends at $i$, then $T[i+1] = T[i] - v$ since no segment starts at $i+1$. □

It now follows immediately from Lemma 7 and Lemma 8, using $M = D$:

**Theorem 4.** *There exists a polynomial-time algorithm that, for any intensity matrix $T$ with maximum difference $D$ between consecutive elements in a row, finds a segmentation $\mathcal{S}$ of $T$ size at most $\alpha \cdot (\log D + 1) OPT$. Here $\alpha \leq \frac{24}{13} \approx 1.846$ in the general case by [3, 4] and $\alpha = 1$ if $h \in O(\log^2 n)$ by Theorem 1.*

For the general case, this improves upon the $2 \cdot (\log D + 1)$ approximation result for the full-matrix problem in [14]. In particular, for $\alpha = \frac{24}{13}$, if $D \leq \left(\frac{h^{13}}{8}\right)^{1/16}$, then to the best of our knowledge, this is the tightest approximation to the segmentation problem with no restriction on the intensity matrix values.

## 5 Experimental Results

In this section, we give experimental results for the following five algorithms:

- XV: The heuristic algorithm of Xia and Verhey [13] extended to the full $m \times n$ case. This algorithm has commonly been used as a benchmark for comparison of new segmentation algorithms [9, 2, 11].
- ALG 1: The $(\log h + 1)$ approximation algorithm of [14].
- ALG 2: The $\frac{3}{2} \cdot (\log_3 h + 1)$ approximation algorithm of Section 3.2.
- ALG 3: The $2(\log D + 1)$ approximation algorithm of [14].
- ALG 4: The $\frac{24}{13} \cdot (\log D + 1)$ approximation algorithm of Section 4, which utilizes our implementations of algorithms from [3–5].

All algorithms were implemented using the Java programming language, using approximately 3600 lines of codes overall.

### 5.1 Data Sets

We used five data sets:

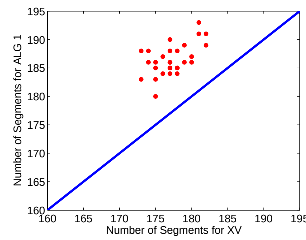
- *Data Set I:* a real-world data set comprised of $70$ clinical intensity matrices obtained from the Department of Radiation Oncology at the University of California at the San Francisco School of Medicine. The type of cancer is unknown and levels are specified in terms of percentages in increments of $20\%$ of some maximum value $v$. For each intensity matrix, we chose $v$ at random from $\{25, ..., 100\}$. This range of values was selected since it is large enough to prevent trivial solutions. Therefore, each matrix contains values from $\{0, \lceil 0.2 \cdot v \rceil, \lceil 0.4 \cdot v \rceil, \lceil 0.6 \cdot v \rceil, \lceil 0.8 \cdot v \rceil, v\}$.
- *Data Set II:* a real-world data set containing a prostate case, a brain case and a head-neck case obtained from the Department of Radiation Oncology at the University of Maryland School of Medicine. This data set consists of $22$ clinical intensity matrices with values specified absolutely.
- *Data Set III:* a synthetic data set of $30$ intensity matrices with values sampled uniformly at random from between $0$ and $10,000$; random matrices have been used previously for performance testing [13].
- *Data Set IV:* a synthetic data set of $20$ intensity matrices. Each matrix is obtained as follows: compute the sum of the pdfs of four bivariate Gaussians generated from two independent standard univariate Gaussian distributions. These are then scaled by $A \cdot 2 \cdot \pi$ where the amplitude $A$ and the centers of the distributions are sampled uniformly at random.Determine the smallest $m \times n$-grid so that the function is less than $1$ outside this grid. Discretize the function, i.e., add as value in the $m \times n$-grid the integer part of the corresponding function value. The choice of "four" Gaussians and the range of the amplitude (we chose 1-80) was made to ensure some peaks and valleys in the intensity matrix, while keeping the matrices reasonably small. We would expect these matrices to have a small $D$-value, since the Gaussian distributions do not rapidly change value.
- *Data Set V:* a synthetic data set of $30$ intensity matrices. For fixed input $D$, $T[r][1] = D$ and for $i = 2, .., n/2$, $T[r][i] = \max\{T[r][i-1] + c, 0\}$ where $c$ is randomly selected from $\{-D, ..., -1, 0, 1, ..., D\}$. For $i = \frac{n}{2} + 1, ..., n$, we set $T[r][i] = T[r][n-i+1]$. Note that these matrices can never have a $D$-value larger than the prescribed bound.

Data Sets IV & V were engineered to possess small $D$ values relative to $h$. Testing on matrices with small $D$ values is pertinent assuming improvements in treatment technology; higher precision MLCs may facilitate treatment plans for more fine-grained intensity matrices. All of our test cases have sizes $m, n$ varying between 20 and 171.

The experiments conducted on Data Sets I, II & III were executed on a machine with a $1$ GHz Pentium CPU and 1GB of RAM; this was also the case for Data Sets IV & V using XV, ALG 1 and ALG 2. For ALG 3 and ALG 4, the experiments involving Data Sets IV & V required more memory and were conducted on a Silicon Graphics Altix 3700 system with $64$ $1.3$ GHz Intel Itanium $2$ CPUs and $192$ GB of memory. We did not use any of the advanced resources of this machine except the increased memory and never utilized more than $2$ GB of RAM in any particular execution. All of the algorithms ran very fast (usually a few seconds) on all trials; the slowest seemed to be ALG 4 and ALG 5, which took up to 45 seconds on some trials in Data Set IV and V. However, evaluating the running time was not the focus of our experiments, and our code was not optimized for it.

| | # trials | XV | ALG 1 | ALG 3 |
|---|---|---|---|---|
| Data Set I | 70 | **41** | 38 | 9 |
| Data Set II | 22 | **22** | 0 | 0 |
| Data Set III | 30 | **30** | 0 | 0 |
| Data Set IV | 20 | 0 | 0 | **20** |
| Data Set V | 30 | 14 | 0 | **18** |

(A)




(B)

**Fig. 2.** (A): The number of trials where XV, ALG 1 and ALG 3 gave the smallest segmentations, respectively. (B): The size of the segmentations of XV *vs* ALG 1 on Data Set III. Each point $(x, y)$ corresponds to a trial where XV used $x$ segments and ALG 1 used $y$ segments. Points above the main diagonal indicate trials where XV outperformed ALG 1. More than one trial may correspond to any $(x, y)$ in the plot; see Tables 2–7 of the appendix for more details.

| | # trials | XV | ALG 1 | ALG 2 | ALG 3 | ALG 4 |
|---|---|---|---|---|---|---|
| Data Set I | 70 | 12 | 17 | **44** | 4 | 4 |
| Data Set II | 22 | 6 | 0 | **16** | 0 | 0 |
| Data Set III | 30 | 3 | 0 | **27** | 0 | 0 |
| Data Set IV | 20 | 0 | 0 | 0 | 3 | **19** |
| Data Set V | 30 | 7 | 0 | 1 | 5 | **24** |

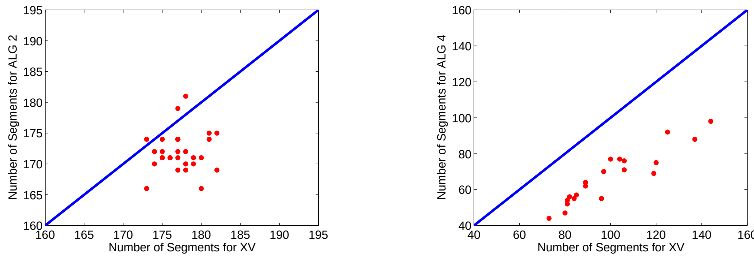
**Table 1.** The number of trials where each of the 5 algorithms achieves the smallest segmentation.

### 5.2 Results

The full tables of results can be found in the appendix. To evaluate them, we focused on two questions: (1) Do the approximation algorithms in [14] give an improvement in practice, i.e., in comparison to the standard benchmark algorithm, XV? (No experimental results were given in that paper.) (2) How do our improved approximation algorithms perform compared to existing algorithms?

With regards to the first question, our experiments show that while ALG 1 comes with approximation guarantees, it shows mediocre performance compared to XV. Figure 2(A) summarizes the number of trials in which each algorithm gave the best segmentations (ties are double counted). XV often produces significantly smaller segmentations, as clearly illustrated in Figure 2(B) for Data Set III. ALG 3 performs better than XV on Data Set IV, which was tailored to suit it well, but shows only a minor performance advantage on Data Set V.

Regarding the second question, our experimental results show that our new approximation algorithms perform considerably better than existing algorithms. Table 1 shows how often each of the five algorithms we considerd achieves the smallest segmentation (ties are double counted.) ALG 2 found the best segmentation in 77/122 trials on Data Sets I–III, whereas ALG 4 found the best segmentation in 43/50 trials on Data Sets IV and V; as seen in Figure 3, these segmentations are often significantly smaller than those produced by XV.




(A) (B)

**Fig. 3.** (A) The size of the segmentation of XV *vs* ALG 2 on Data Set III. (B) The size of the segmentation of XV *vs* ALG 4 on Data Set IV.

# 6 Conclusion

We provided new approximation algorithms for the full-matrix segmentation problem. We first showed that the single-row segmentation problem is fixed-parameter tractable in the largest value of the intensity matrix. Using this yields provably good approximate segmentations for the full matrix, after suitably splitting either the intensity matrix or approximate segmentations of its rows according to some base-$b$ representation. Finally, our experimental results demonstrate that our theoretical improvements yield new algorithms that, in both the $O(\log h)$ and $O(\log D)$ cases, significantly outperform previous approximation algorithms in practice.

It may be of interest to explore the case of $b \geq 4$ as a base further. Can we solve the matrix segmentation problem optimally if all values are in $\{0, 1, 2, 3\}$? And does this lead to better approximation algorithms? Are further heuristic improvements possible, such that empirical performance in practically relevant cases is increased, while maintaining desirable theoretical approximation guarantees?

# References

1. D. Baatar and H. W. Hamacher. New LP model for Multileaf Collimators in Radiation Therapy. Contribution to the Conference ORP3, Univ. Kaiserslautern, 2003.
2. D. Baatar, W. Hamacher, M. Ehrgott and G. J. Woeginger. Decomposition of Integer Matrices and Multileaf Collimator Sequencing. Discrete Applied Mathematics, 152, 6-34, 2005.
3. N. Bansal, D. Coppersmith, B. Schieber. Minimizing Setup and Beam-On Times in Radiation Therapy. In Proceedings of APPROX, 2006.
4. N. Bansal, D. Chen, D. Coppersmith, Xiaobo S. Hu, Shuang Luan, Ewa Misiolek, B. Schieber and Chao Wang. Shape Rectangularization Problems in Intensity-Modulated Radiation Therapy. In submission, 2008.
5. Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph Naor and Baruch Schieber. A Unified Approach to Approximating Resource Allocation and Scheduling. In Proceedings of STOC, 2000.
6. D.Z. Chen, X.S. Hu, S. Luan, S.A. Naqvi, C. Wang, and C.X. Yu. Generalized Geometric Approaches for Leaf Sequencing Problems in Radiation Therapy. Proc. 15th Intl. Symp. on Algorithms and Computation (ISAAC), 271-281, 2004.
7. M. J. Collins, D. Kempe, J. Saia, and M. Young. Non-Negative Integral Subset Representations of Integer Sets. Information Processing Letters 101(3), 129-133, 2007.
8. C. Cotrutz and L. Xing. Segment-Based Dose Optimization Using a Genetic Algorithm. Physics in Medicine and Biology, 48, 2987-2998, 2003.

9. J. Dai and Y. Zhu. Minimizing the Number of Segments in a Delivery Sequence for Intensity-Modulated Radiation Therapy with a Multileaf Collimator. Medical Physics, 28(10), 2113-2120, 2001.
10. W. Pribitkin. Simple Upper Bounds for Partition Functions. The Ramanujan Journal. Online at: http://www.springerlink.com/content/d65348128235x7m1/
11. K. Engel. A New Algorithm for Optimal Multileaf Collimator Field Segmentation. Discrete Applied Mathematics, 152, 35-51, 2005.
12. S. Luan, C. Wang, D.Z. Chen, X. S. Hu, S.A. Naqvi, C.L. Lee, and C.X. Yu. A new MLC segmentation algorithm/software for step-and-shoot IMRT delivery. Medical Physics, 31, 695-707, 2004.
13. P. Xia and L. Verhey. Multileaf Collimator Leaf Sequencing Algorithm for Intensity Modulated Beams with Multiple Static Segments. Medical Physics, 1998.
14. S. Luan, J. Saia and M. Young. Approximation Algorithms for Minimizing Segments in Radiation Therapy. Information Processing Letters, 101, 239-244, 2007.

## A Proofs

**Lemma 1.** *Every single-row problem has an optimal segmentation $\mathcal{S}$ that is compact, i.e., any two segments of $\mathcal{S}$ begin at different indices and end at different indices.*

*Proof.* Start with an arbitrary optimal segmentation $\mathcal{S}$; we can argue how to modify $\mathcal{S}$ to obtain a compact segmentation of the same size. Let $i$ be the smallest index such that two segments $S, S'$ of $\mathcal{S}$ begin at index $i$. Say $S$ and $S'$ have non-zero value $a$ and $a'$ and end at index $j$ and $j'$, respectively. If $j = j'$, then the two segments could be combined into one to give a smaller segmentation, a contradiction. So $j \neq j'$, say $j < j'$.

Define two new segments $S''$ and $S'''$ as follows. Segment $S''$ begins at $i$, ends at $j$ and has value $a + a'$. Segment $S'''$ begins at $j + 1$, ends at $j'$, and has value $a'$. Clearly $S + S' = S'' + S'''$, so $\mathcal{S}' = \mathcal{S} - \{S, S'\} \cup \{S'', S'''\}$ is also an optimal segmentation, and has fewer segments that start at $i$. Iterate until only one segment starts at $i$, then iterate with all larger values where multiple segments start. (Note that all new segments in $\mathcal{S}'$ start at $i$ or later, so this eliminates all coinciding start-indices.) Then similarly eliminate coinciding end-indices, starting at the largest one where they occur. □

**Lemma 2** *For $i > 1$, $f(i, \phi_i) = \min\limits_{\phi_{i-1} \in \Phi_{i-1}(\phi_i)} \{f(i-1, \phi_{i-1}) + ||\phi_i - \phi_{i-1}||\}$*

*Proof.* To prove "$\leq$", let $\phi_{i-1} \in \Phi_{i-1}(\phi_i)$ be a partition of $T[i]$ that achieves the minimum on the right-hand side. Let $\mathcal{S}_{i-1}$ be an almost-compact segmentation that achieves $f(i-1, \phi_{i-1}$, i.e., it is a partition of $T[1..i-1]$ with signature $\phi_{i-1}$ and size $f(i-1, \phi_{i-1})$. Define a segmentation $\mathcal{S}_i$ of $T[1..i]$ as follows. Every segment of $\mathcal{S}_{i-1}$ that ends before index $i-1$ is added to $\mathcal{S}_i$ as is. For each value in $\phi_{i-1} - \phi_i$, there must be a segment in $\mathcal{S}_{i-1}$ that ends at index $i-1$; add this segment to $\mathcal{S}_i$ and let it end at $i-1$ (i.e., set its $i$th entry to be 0). For each value in $\phi_{i-1} \cap \phi_i$, there must be a segment in $\mathcal{S}_{i-1}$ that ends at index $i-1$; add this segment to $\mathcal{S}_i$ and extend it to $i$ (i.e., set its $i$th entry to be the same as its $(i-1)$st entry.) For each value in $\phi_i - \phi_{i-1}$, define a new segment in $\mathcal{S}_i$ that starts at $i$ and has that value at index $i$. One easily verifies that $\mathcal{S}_i$ has signature $\phi_i$, and therefore is a segmentation of $T[1..i]$, since $\phi_i$ is a partition of $T[i]$. We can convert it to an almost-compact segmentation as in the proof of Lemma 1. Also, $|\mathcal{S}_i| = |\mathcal{S}_{i-1}| + ||\phi_i - \phi_{i-1}||$, which proves the result. □

**Lemma 4** *Any single row with values in $\{0, 1, 2\}$ has a segmentation such that the number of 1-segments is at most $\frac{1}{2}$#markers, and the number of 2-segments is at most $\frac{1}{4}$#markers$+\frac{1}{2}$.*

*Proof.* We prove this by repeatedly identifying a subsequence of the row for which we can add a few segments and remove many markers, where "remove" means that if we subtracted the segments from the target row, we would have fewer markers. To identify subsequences of the row, we again use regular expression notations.

1. As long as there exists a subsequence of the form $12^+1$, apply a 1-segment at the subsequence of 2s. This removes 2 markers, adds one 1-segment, and no 2-segment.
2. As long as there exists a subsequence of the form $01^+0$, apply a 1-segment at the subsequence of 1s. This removes 2 markers, adds one 1-segment, and no 2-segment.
3. As long as there exists a subsequence of the form $02^+1^+2^+0$, apply a 2-segment at the first subsequence of 2s, then two 1-segments to remove the remaining $1^+2^+$. This removes 4 markers, adds two 1-segments, and one 2-segment.

4. As long as there exist two subsequences of the form $02^+1^+0$ or $01^+2^+0$, apply one 1-segment to one subsequence of 2s, and one 2-segment to the other subsequence of 2s, then apply two 1-segments to the two remaining sequences of 1s. This removes 6 markers, adds three 1-segments and one 2-segment.
5. As long as there exist two subsequences of the form $02^+0$, apply one 2-segment to one of them, and two 1-segments to the other. This removes 4 markers, adds two 1-segments and one 2-segment.
6. If there exists one subsequence of the form $02^+1^+0$ or $01^+2^+0$, and one subsequence of the form $02^+0$, apply one 2-segment to the subsequence $02^+0$, and two one 1-segments to the other subsequence. This removes 5 markers, adds two 1-segments and one 2-segment.

In all the above cases, we have removed at least 2 markers per 1-segment and at least 4 markers per 2-segment. Thus, counting only segments created and markers removed thus far, we have at most $\frac{1}{2}\#$markers 1-segments and $\frac{1}{4}\#$markers 2-segments. All that remains to do is to consider any markers that are remaining.

We argue that in fact at most three markers are left. Let $0(1+2)^+0$ be a subsequence that has markers in it. Assume first the leftmost non-zero is a 1. Then the subsequence must contain a 2 somewhere (otherwise we're in case (2)), so it has the form $01^+2^+(1+2)^+0$. But after the 2s, no 1 can follow (otherwise we're in case (1)), so this subsequence has the form $01^+2^+0$. Likewise, if the last non-zero is 1, then the subsequence has the form $02^+1^+0$. If the first and last non-zero are 2, then the subsequence has the form $02^+0$ (otherwise we're in case (1) or (3)).

If we had two subsequences $0(1+2)^+0$, then each would have the form $01^+2^+0$ or $02^+1^+0$ or $02^+0$, and we would be in case (4),(5) or (6). So there is only one of them, and it has at most three markers.

We can now eliminate either three remaining markers with a 1-segment and a 2-segment, or two remaining markers with a 2-segment; either way the bound holds. □

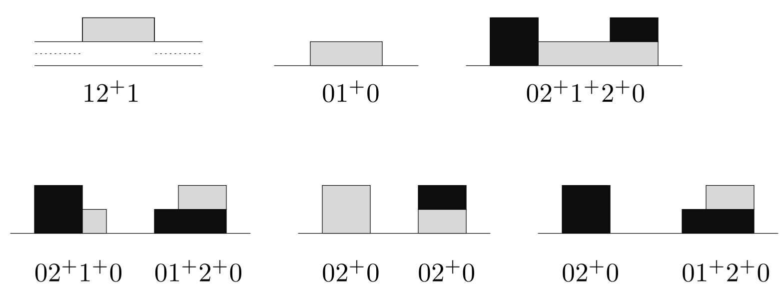



**Fig. 4.** A segmentation where the number of segments is bounded by markers.

## B Experimental results

Below are Tables 2-7 from Section 5 which contain the results for each trial of our experimental evaluation.

| Trial | m | n | # markers | $h$ | $D$ | XV | ALG 1 | ALG 2 | ALG 3 | ALG 4 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 27 | 21 | 133 | 45 | 45 | 17 | 20 | 8 | 24 | 24 |
| 2 | 27 | 21 | 132 | 62 | 62 | 21 | 22 | 21 | 23 | 23 |
| 3 | 27 | 21 | 121 | 80 | 80 | 19 | 11 | 25 | 15 | 15 |
| 4 | 27 | 21 | 128 | 93 | 93 | 25 | 26 | 22 | 31 | 35 |
| 5 | 27 | 21 | 143 | 74 | 74 | 25 | 24 | 21 | 35 | 35 |
| 6 | 27 | 21 | 134 | 41 | 41 | 16 | 14 | 18 | 15 | 22 |
| 7 | 27 | 21 | 145 | 94 | 94 | 29 | 29 | 24 | 33 | 34 |
| 8 | 27 | 21 | 151 | 92 | 92 | 25 | 27 | 20 | 30 | 31 |
| 9 | 27 | 21 | 127 | 42 | 42 | 18 | 15 | 15 | 19 | 20 |
| 10 | 27 | 21 | 224 | 90 | 90 | 35 | 35 | 29 | 46 | 46 |
| 11 | 27 | 21 | 209 | 36 | 36 | 26 | 24 | 25 | 26 | 26 |
| 12 | 27 | 21 | 180 | 82 | 82 | 27 | 27 | 14 | 32 | 32 |
| 13 | 27 | 21 | 163 | 61 | 61 | 20 | 17 | 19 | 19 | 19 |
| 14 | 27 | 21 | 187 | 28 | 28 | 19 | 22 | 13 | 24 | 24 |
| 15 | 27 | 21 | 177 | 75 | 75 | 27 | 24 | 20 | 25 | 25 |
| 16 | 27 | 21 | 147 | 45 | 45 | 17 | 18 | 17 | 19 | 19 |
| 17 | 27 | 21 | 191 | 94 | 94 | 27 | 27 | 24 | 30 | 30 |
| 18 | 27 | 21 | 200 | 47 | 47 | 22 | 23 | 20 | 22 | 22 |
| 19 | 27 | 21 | 192 | 42 | 42 | 25 | 24 | 24 | 20 | 24 |
| 20 | 27 | 21 | 156 | 85 | 85 | 19 | 18 | 24 | 24 | 24 |
| 21 | 27 | 21 | 136 | 53 | 53 | 20 | 21 | 19 | 27 | 28 |
| 22 | 27 | 21 | 175 | 68 | 68 | 26 | 28 | 22 | 37 | 36 |
| 23 | 27 | 21 | 169 | 84 | 84 | 27 | 22 | 28 | 24 | 25 |
| 24 | 27 | 21 | 129 | 69 | 69 | 22 | 20 | 16 | 23 | 23 |
| 25 | 27 | 21 | 175 | 38 | 38 | 19 | 21 | 23 | 27 | 27 |
| 26 | 27 | 21 | 193 | 84 | 84 | 28 | 26 | 31 | 25 | 25 |
| 27 | 27 | 21 | 181 | 51 | 51 | 26 | 21 | 24 | 24 | 25 |
| 28 | 27 | 21 | 188 | 71 | 71 | 32 | 36 | 29 | 33 | 33 |
| 29 | 27 | 21 | 146 | 43 | 43 | 21 | 21 | 18 | 27 | 27 |
| 30 | 27 | 21 | 92 | 31 | 31 | 13 | 13 | 12 | 15 | 15 |
| 31 | 27 | 21 | 157 | 69 | 69 | 34 | 35 | 28 | 37 | 37 |
| 32 | 27 | 21 | 174 | 31 | 31 | 22 | 22 | 20 | 28 | 28 |
| 33 | 27 | 21 | 142 | 55 | 55 | 17 | 19 | 14 | 21 | 21 |
| 34 | 27 | 21 | 171 | 70 | 70 | 33 | 33 | 29 | 38 | 38 |
| 35 | 27 | 21 | 121 | 38 | 38 | 18 | 21 | 21 | 23 | 23 |

**Table 2.** The experimental trials 1-35 using Data Set I with the best result underscored.

| Trial | m | n | # markers | $h$ | $D$ | XV | ALG 1 | ALG 2 | ALG 3 | ALG 4 |
|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 27 | 21 | 136 | 87 | 87 | 24 | 25 | 19 | 33 | 33 |
| 37 | 27 | 21 | 181 | 86 | 86 | 24 | 22 | 22 | 22 | 21 |
| 38 | 27 | 21 | 224 | 60 | 60 | 34 | 22 | 24 | 25 | 24 |
| 39 | 27 | 21 | 178 | 50 | 50 | 22 | 19 | 20 | 25 | 23 |
| 40 | 27 | 21 | 244 | 90 | 90 | 29 | 24 | 20 | 40 | 28 |
| 41 | 27 | 21 | 255 | 83 | 83 | 36 | 26 | 38 | 30 | 28 |
| 42 | 27 | 21 | 226 | 65 | 65 | 25 | 28 | 26 | 34 | 34 |
| 43 | 27 | 21 | 174 | 82 | 82 | 21 | 16 | 26 | 18 | 17 |
| 44 | 27 | 21 | 173 | 67 | 67 | 25 | 27 | 20 | 29 | 29 |
| 45 | 27 | 21 | 207 | 35 | 35 | 26 | 29 | 26 | 36 | 36 |
| 46 | 27 | 21 | 32 | 29 | 29 | 12 | 10 | 11 | 13 | 13 |
| 47 | 27 | 21 | 31 | 35 | 35 | 10 | 12 | 11 | 18 | 16 |
| 48 | 27 | 21 | 40 | 54 | 54 | 15 | 17 | 14 | 16 | 16 |
| 49 | 27 | 21 | 32 | 73 | 73 | 12 | 12 | 14 | 22 | 22 |
| 50 | 27 | 21 | 44 | 31 | 31 | 11 | 11 | 12 | 14 | 14 |
| 51 | 27 | 21 | 42 | 84 | 84 | 18 | 17 | 19 | 21 | 21 |
| 52 | 27 | 21 | 39 | 57 | 57 | 14 | 15 | 14 | 17 | 16 |
| 53 | 27 | 21 | 30 | 38 | 38 | 10 | 11 | 13 | 17 | 16 |
| 54 | 27 | 21 | 41 | 71 | 71 | 19 | 18 | 15 | 24 | 26 |
| 55 | 27 | 21 | 137 | 55 | 55 | 17 | 19 | 15 | 23 | 23 |
| 56 | 27 | 21 | 109 | 45 | 45 | 21 | 20 | 16 | 20 | 20 |
| 57 | 27 | 21 | 94 | 70 | 70 | 18 | 17 | 14 | 17 | 17 |
| 58 | 27 | 21 | 105 | 63 | 63 | 19 | 19 | 16 | 21 | 21 |
| 59 | 27 | 21 | 96 | 80 | 80 | 20 | 19 | 11 | 23 | 23 |
| 60 | 27 | 21 | 58 | 65 | 65 | 10 | 7 | 10 | 6 | 6 |
| 61 | 27 | 21 | 122 | 53 | 53 | 24 | 17 | 14 | 17 | 17 |
| 62 | 27 | 21 | 130 | 89 | 89 | 24 | 25 | 19 | 33 | 33 |
| 63 | 27 | 21 | 118 | 98 | 98 | 18 | 14 | 22 | 14 | 14 |
| 64 | 27 | 21 | 195 | 28 | 28 | 21 | 22 | 25 | 27 | 28 |
| 65 | 27 | 21 | 136 | 21 | 21 | 14 | 15 | 15 | 16 | 16 |
| 66 | 27 | 21 | 77 | 71 | 71 | 11 | 12 | 10 | 13 | 13 |
| 67 | 27 | 21 | 167 | 74 | 74 | 26 | 29 | 20 | 41 | 37 |
| 68 | 27 | 21 | 130 | 90 | 90 | 21 | 22 | 14 | 22 | 22 |
| 69 | 27 | 21 | 99 | 24 | 24 | 12 | 12 | 8 | 13 | 11 |
| 70 | 27 | 21 | 133 | 54 | 54 | 21 | 21 | 14 | 25 | 25 |
| Total | | | | | | 12 | 17 | 44 | 4 | 4 |

**Table 3.** The experimental trials 36-70 using Data Set I with the best result underscored.

| Trial | m | n | # markers | $h$ | $D$ | XV | ALG 1 | ALG 2 | ALG 3 | ALG 4 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 20 | 20 | 209 | 10000000 | 8805586 | 114 | 124 | 108 | 203 | 203 |
| 2 | 20 | 21 | 219 | 10000000 | 8399071 | 117 | 129 | 108 | 216 | 215 |
| 3 | 20 | 20 | 210 | 10000000 | 9340255 | 114 | 123 | 108 | 212 | 193 |
| 4 | 20 | 19 | 198 | 10000000 | 9909173 | 100 | 106 | 97 | 178 | 178 |
| 5 | 20 | 21 | 229 | 10000000 | 9722947 | 121 | 127 | 113 | 211 | 209 |
| 6 | 20 | 21 | 231 | 10000000 | 8504569 | 119 | 129 | 118 | 214 | 218 |
| 7 | 20 | 19 | 193 | 10000000 | 9262494 | 100 | 108 | 102 | 184 | 188 |
| 8 | 28 | 31 | 502 | 10000000 | 9275133 | 182 | 192 | 185 | 338 | 337 |
| 9 | 28 | 29 | 475 | 10000000 | 7856988 | 163 | 176 | 164 | 307 | 304 |
| 10 | 28 | 37 | 568 | 10000000 | 9056646 | 211 | 221 | 206 | 368 | 388 |
| 11 | 27 | 28 | 449 | 10000000 | 8102019 | 162 | 171 | 159 | 306 | 303 |
| 12 | 27 | 31 | 493 | 10000000 | 7928634 | 180 | 189 | 171 | 322 | 310 |
| 13 | 27 | 38 | 610 | 10000000 | 6831687 | 220 | 234 | 219 | 414 | 408 |
| 14 | 28 | 42 | 635 | 10000000 | 9998558 | 234 | 247 | 231 | 425 | 428 |
| 15 | 26 | 28 | 379 | 10000000 | 9959856 | 147 | 156 | 154 | 276 | 273 |
| 16 | 25 | 24 | 313 | 10000000 | 8218883 | 134 | 140 | 123 | 233 | 233 |
| 17 | 25 | 27 | 340 | 10000000 | 9996029 | 138 | 149 | 143 | 253 | 268 |
| 18 | 24 | 28 | 366 | 10000000 | 8915037 | 151 | 158 | 145 | 287 | 296 |
| 19 | 24 | 25 | 352 | 10000000 | 6870038 | 142 | 153 | 138 | 266 | 273 |
| 20 | 25 | 24 | 330 | 10000000 | 6226761 | 133 | 141 | 135 | 243 | 236 |
| 21 | 26 | 28 | 385 | 10000000 | 9698595 | 160 | 173 | 154 | 302 | 292 |
| 22 | 26 | 25 | 345 | 10000000 | 8893105 | 141 | 143 | 137 | 248 | 253 |
| Total | | | | | | 6 | 0 | 16 | 0 | 0 |

**Table 4.** Experimental trials using Data Set II with the best result underscored.

| Trial | m | n | # markers | $h$ | $D$ | XV | ALG 1 | ALG 2 | ALG 3 | ALG 4 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 40 | 40 | 1640 | 9992 | 9860 | 174 | 186 | 170 | 301 | 323 |
| 2 | 40 | 40 | 1640 | 9998 | 9866 | 179 | 186 | 171 | 304 | 321 |
| 3 | 40 | 40 | 1640 | 10000 | 10000 | 176 | 184 | 171 | 302 | 309 |
| 4 | 40 | 40 | 1639 | 9991 | 9969 | 181 | 193 | 175 | 310 | 308 |
| 5 | 40 | 40 | 1640 | 9997 | 9953 | 177 | 188 | 172 | 303 | 303 |
| 6 | 40 | 40 | 1640 | 9982 | 9804 | 177 | 185 | 172 | 313 | 311 |
| 7 | 40 | 40 | 1640 | 9980 | 9921 | 175 | 183 | 172 | 300 | 307 |
| 8 | 40 | 40 | 1639 | 9996 | 9855 | 177 | 186 | 174 | 303 | 312 |
| 9 | 40 | 40 | 1640 | 9993 | 9798 | 175 | 180 | 171 | 313 | 320 |
| 10 | 40 | 40 | 1639 | 9986 | 9844 | 176 | 187 | 171 | 302 | 302 |
| 11 | 40 | 40 | 1640 | 9995 | 9946 | 177 | 186 | 179 | 315 | 310 |
| 12 | 40 | 40 | 1640 | 10000 | 9898 | 180 | 186 | 171 | 304 | 313 |
| 13 | 40 | 40 | 1640 | 9999 | 9769 | 177 | 184 | 174 | 299 | 310 |
| 14 | 40 | 40 | 1640 | 9987 | 9898 | 175 | 186 | 174 | 303 | 308 |
| 15 | 40 | 40 | 1640 | 9997 | 9943 | 178 | 185 | 172 | 304 | 314 |
| 16 | 40 | 40 | 1640 | 9997 | 9979 | 182 | 189 | 169 | 300 | 315 |
| 17 | 40 | 40 | 1640 | 9998 | 9971 | 177 | 185 | 174 | 308 | 318 |
| 18 | 40 | 40 | 1640 | 9995 | 9977 | 177 | 185 | 169 | 308 | 312 |
| 19 | 40 | 40 | 1640 | 9999 | 9940 | 180 | 187 | 166 | 297 | 310 |
| 20 | 40 | 40 | 1640 | 9994 | 9912 | 173 | 183 | 166 | 301 | 311 |
| 21 | 40 | 40 | 1640 | 9996 | 9996 | 182 | 191 | 175 | 312 | 307 |
| 22 | 40 | 40 | 1640 | 9999 | 9893 | 178 | 188 | 170 | 307 | 309 |
| 23 | 40 | 40 | 1640 | 9996 | 9969 | 178 | 185 | 181 | 301 | 311 |
| 24 | 40 | 40 | 1640 | 9986 | 9898 | 175 | 185 | 171 | 294 | 311 |
| 25 | 40 | 40 | 1640 | 9975 | 9929 | 181 | 191 | 174 | 305 | 316 |
| 26 | 40 | 40 | 1640 | 10000 | 9983 | 178 | 184 | 169 | 306 | 315 |
| 27 | 40 | 40 | 1640 | 9995 | 9907 | 173 | 188 | 174 | 298 | 311 |
| 28 | 40 | 40 | 1640 | 9976 | 9921 | 174 | 188 | 172 | 300 | 316 |
| 29 | 40 | 40 | 1640 | 9989 | 9928 | 177 | 190 | 171 | 301 | 309 |
| 30 | 40 | 40 | 1640 | 9998 | 9988 | 179 | 189 | 170 | 309 | 308 |
| Total | | | | | | 3 | 0 | 27 | 0 | 0 |

**Table 5.** Experimental trials using Data Set III with the best result underscored.

| Trial | m | n | # markers | $h$ | $D$ | XV | ALG 1 | ALG 2 | ALG 3 | ALG 4 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 151 | 146 | 9115 | 50 | 3 | 119 | 125 | 118 | 79 | 69 |
| 2 | 129 | 171 | 8825 | 73 | 3 | 137 | 142 | 137 | 100 | 88 |
| 3 | 149 | 106 | 6740 | 76 | 4 | 89 | 94 | 88 | 70 | 64 |
| 4 | 117 | 123 | 6540 | 87 | 5 | 106 | 112 | 113 | 84 | 76 |
| 5 | 165 | 148 | 8946 | 77 | 4 | 97 | 101 | 99 | 77 | 70 |
| 6 | 170 | 136 | 6746 | 73 | 5 | 82 | 86 | 84 | 61 | 56 |
| 7 | 135 | 158 | 8826 | 75 | 4 | 120 | 129 | 128 | 97 | 75 |
| 8 | 113 | 108 | 6106 | 84 | 5 | 100 | 104 | 100 | 75 | 77 |
| 9 | 109 | 148 | 6715 | 84 | 5 | 144 | 150 | 140 | 108 | 98 |
| 10 | 162 | 110 | 7213 | 66 | 4 | 81 | 84 | 81 | 66 | 54 |
| 11 | 109 | 135 | 4412 | 60 | 3 | 96 | 101 | 90 | 65 | 55 |
| 12 | 115 | 125 | 3452 | 46 | 3 | 80 | 86 | 83 | 55 | 47 |
| 13 | 125 | 134 | 5597 | 66 | 4 | 89 | 93 | 86 | 62 | 62 |
| 14 | 105 | 132 | 5287 | 150 | 8 | 125 | 128 | 122 | 103 | 92 |
| 15 | 152 | 116 | 7322 | 74 | 4 | 85 | 88 | 87 | 67 | 57 |
| 16 | 121 | 140 | 5701 | 85 | 5 | 104 | 108 | 106 | 77 | 77 |
| 17 | 165 | 125 | 7457 | 46 | 3 | 73 | 76 | 72 | 53 | 44 |
| 18 | 131 | 114 | 5146 | 72 | 4 | 81 | 85 | 84 | 70 | 52 |
| 19 | 131 | 127 | 5023 | 77 | 5 | 106 | 110 | 106 | 73 | 71 |
| 20 | 119 | 153 | 5164 | 58 | 4 | 84 | 86 | 81 | 63 | 55 |
| Total | | | | | | 0 | 0 | 0 | 3 | 19 |

**Table 6.** Experimental trials using Data Set IV with the best result underscored.

| Trial | m | n | # markers | $h$ | $D$ | XV | ALG 1 | ALG 2 | ALG 3 | ALG 4 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 60 | 60 | 3052 | 17 | 1 | 50 | 76 | 55 | 29 | 29 |
| 2 | 60 | 60 | 2976 | 12 | 1 | 46 | 66 | 55 | 29 | 29 |
| 3 | 60 | 60 | 3026 | 12 | 1 | 47 | 69 | 57 | 29 | 29 |
| 4 | 60 | 60 | 3132 | 18 | 2 | 54 | 74 | 71 | 53 | 51 |
| 5 | 60 | 60 | 3120 | 22 | 2 | 61 | 80 | 66 | 51 | 48 |
| 6 | 60 | 60 | 3190 | 20 | 2 | 62 | 74 | 71 | 52 | 48 |
| 7 | 60 | 60 | 3202 | 32 | 3 | 69 | 93 | 67 | 58 | 55 |
| 8 | 60 | 60 | 3232 | 28 | 3 | 75 | 94 | 80 | 60 | 57 |
| 9 | 60 | 60 | 3220 | 29 | 3 | 69 | 91 | 69 | 62 | 55 |
| 10 | 60 | 60 | 3214 | 41 | 4 | 68 | 87 | 79 | 80 | 69 |
| 11 | 60 | 60 | 3212 | 43 | 4 | 72 | 83 | 74 | 77 | 72 |
| 12 | 60 | 60 | 3262 | 34 | 4 | 73 | 93 | 77 | 69 | 68 |
| 13 | 60 | 60 | 3230 | 51 | 5 | 77 | 98 | 86 | 76 | 67 |
| 14 | 60 | 60 | 3288 | 45 | 5 | 80 | 94 | 94 | 78 | 70 |
| 15 | 60 | 60 | 3292 | 46 | 5 | 84 | 96 | 87 | 74 | 73 |
| 16 | 60 | 60 | 3342 | 51 | 6 | 83 | 102 | 90 | 88 | 80 |
| 17 | 60 | 60 | 3270 | 44 | 6 | 84 | 102 | 86 | 84 | 84 |
| 18 | 60 | 60 | 3122 | 54 | 6 | 81 | 92 | 91 | 81 | 74 |
| 19 | 60 | 60 | 3278 | 63 | 7 | 87 | 107 | 91 | 91 | 84 |
| 20 | 60 | 60 | 3348 | 66 | 7 | 86 | 103 | 91 | 89 | 85 |
| 21 | 60 | 60 | 3280 | 53 | 7 | 87 | 101 | 90 | 85 | 78 |
| 22 | 60 | 60 | 3210 | 73 | 8 | 89 | 104 | 92 | 85 | 90 |
| 23 | 60 | 60 | 3292 | 82 | 8 | 91 | 104 | 91 | 95 | 82 |
| 24 | 60 | 60 | 3322 | 71 | 8 | 93 | 110 | 93 | 89 | 85 |
| 25 | 60 | 60 | 3262 | 88 | 9 | 89 | 105 | 95 | 97 | 98 |
| 26 | 60 | 60 | 3284 | 76 | 9 | 94 | 113 | 87 | 105 | 87 |
| 27 | 60 | 60 | 3314 | 83 | 9 | 92 | 118 | 93 | 98 | 95 |
| 28 | 60 | 60 | 3336 | 86 | 10 | 94 | 119 | 108 | 103 | 99 |
| 29 | 60 | 60 | 3386 | 93 | 10 | 97 | 117 | 104 | 107 | 93 |
| 30 | 60 | 60 | 3288 | 90 | 10 | 89 | 103 | 93 | 99 | 101 |
| Total | | | | | | 7 | 0 | 1 | 5 | 24 |

**Table 7.** Experimental trials using Data Set V with the best result underscored.