Fast and Space-Efficient Location of Heavy or Dense Segments in Run-Length Encoded Sequences

Ronald I. Greenberg
Loyola University of Chicago
rig@cs.luc.edu

Outline

- Problem Definition
- Summary of Results
- Some Algorithmic Details
- Conclusion

The Problem

- Several variations — details upcoming
- Given sequence S of n runs where ith run has weight w_i and length $l_i \geq 0$.
- With *atomic* or *unbreakable* runs, find *optimal* segments $S(i, j)$ comprised of runs i through j to maximize segment *value*.

Two Possibilities for Segment Value

- $weight(i, j)$ — finding heaviest segment
- $density(i, j)$ — finding densest segment

where

$$\begin{aligned} weight(i,j) &= \sum_{k=i}^{j} w_k \\ length(i,j) &= \sum_{k=i}^{j} l_k \\ density(i,j) &= weight(i,j)/length(i,j) \end{aligned}$$

Length Constraints

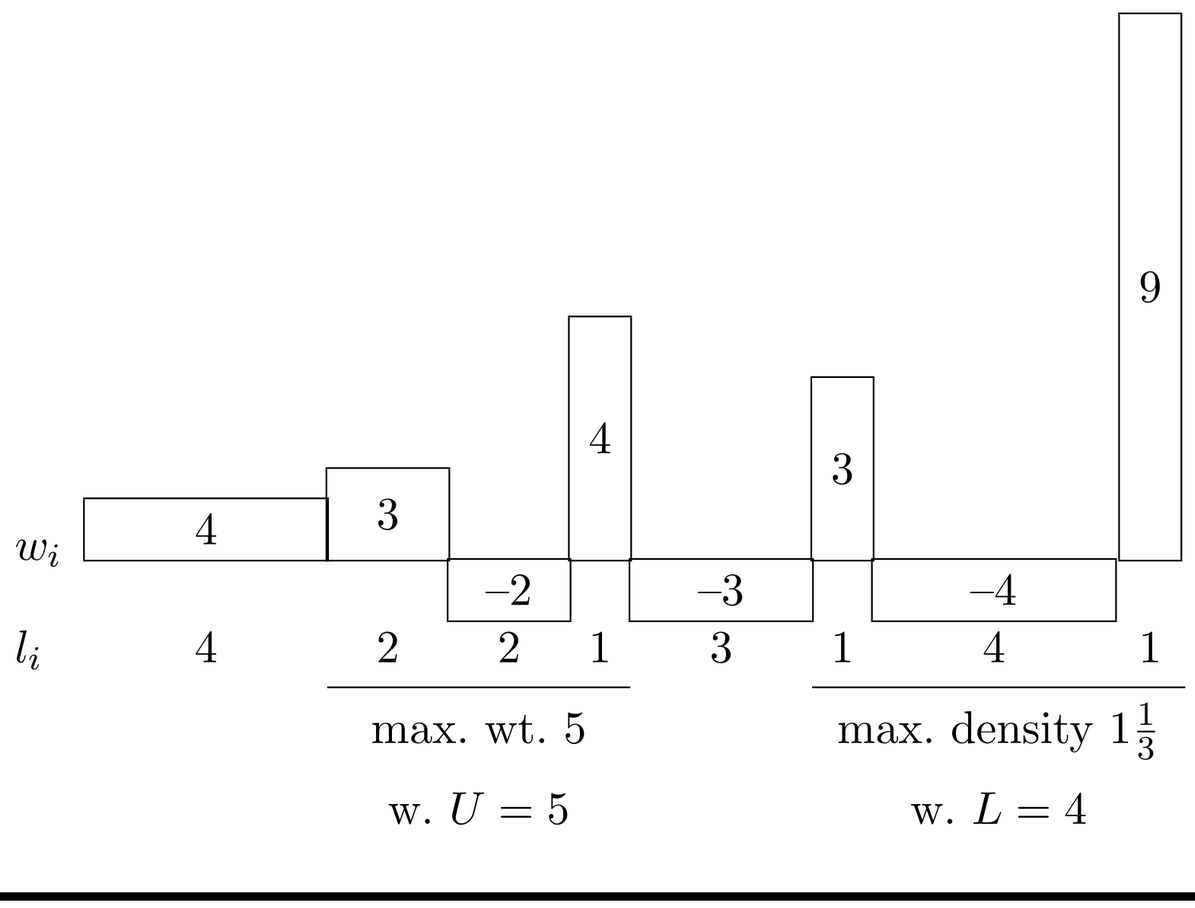
For a segment $S(i, j)$ to be considered optimal, we may require

- $length(i, j) \geq L$ (an L-constraint)

and/or

- $length(i, j) \leq U$ (a U-constraint)

An Example (Atomic Runs)



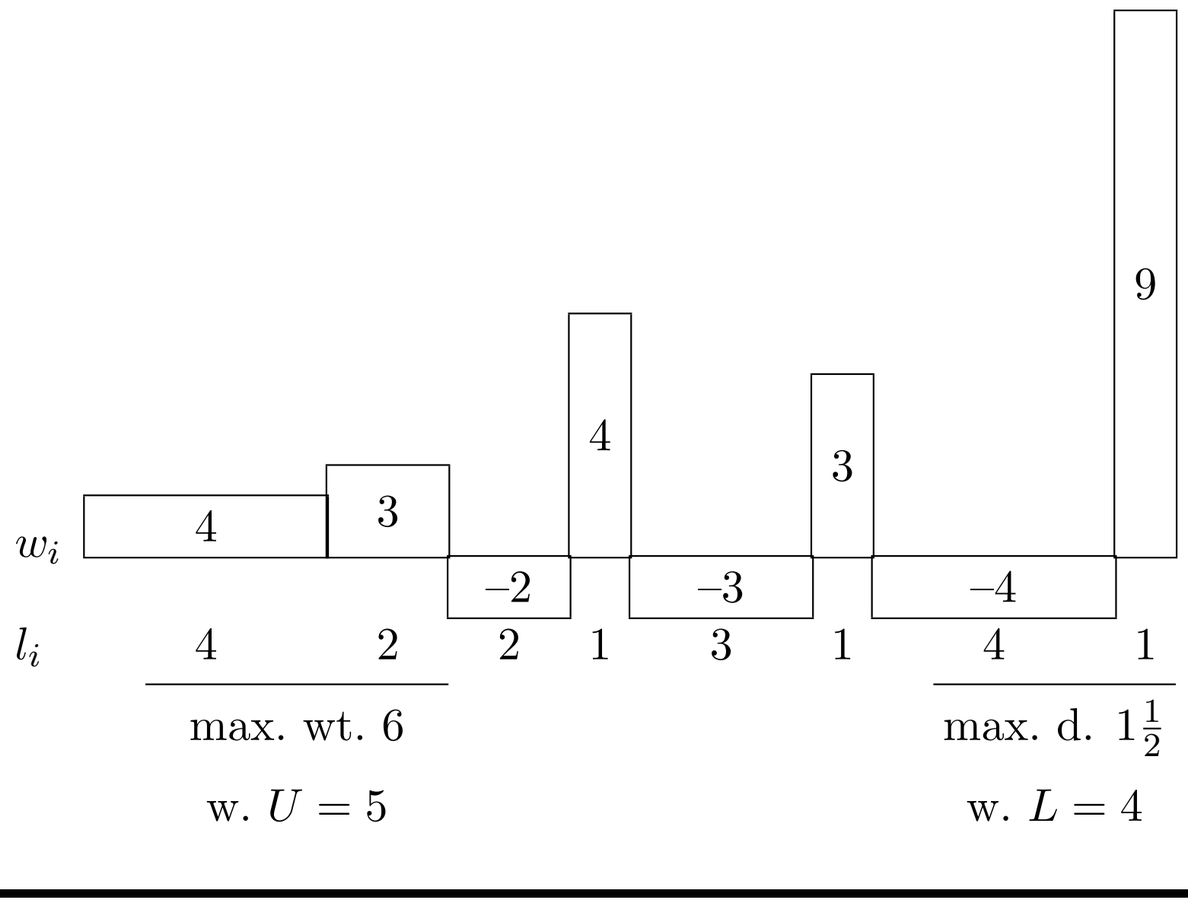
Atomic Versus Non-Atomic Runs

- Prior works considered only atomic runs and usually only the "uniform" model ($l_i = 1 \forall i$).
- One advantage of non-uniformity and breakable runs is that we can work with sequences compressed via run-length encoding, e.g.,:

	A	C	C	G	G	A	A	A	A	T	T		A	C^2	G^2	A^4	T^2
l_i	1	1	1	1	1	1	1	1	1	1	1	$\Rightarrow$	1	2	2	4	2
w_i	1	2	2	3	3	1	1	1	1	4	4		1	4	6	4	8

Ink colored bars between runs to make easier to see.

An Example (Non-Atomic Runs)



Some Application Areas

- image processing [e.g., Bentley 1984/2000] (also 2-D version)
- image processing or correlations among data [e.g., Takaoka 2001, Fukuda-Morimoto-Morishita-Tokuyama 1996, Agrawal-Imielinski-Swami 1993] (2-D generalization of the problem)
- biological sequence analysis, e.g., find GC-rich regions of a DNA sequence [e.g., Lin-Jiang-Chao 2002, Huang 1994, Hannenhalli-Levy 2001, Nekrutenko-Li 2002, Larsen-Gundersen-Lopez-Prydz 1992, Hardison et. al. 1991, Garden-Frommer 1987]

Bentley dates are date of programming pearls CACM column and date of newest edition of book. Biological sequence analysis references start with two more algorithmically oriented papers and then more biology-oriented papers especially oriented to GC content.

Finding Heaviest Segments — Results

	atomic $l_i = 1$	$l_i \geq 0$
unconstrained	$O(n)$ time, $O(1)$ space [K]	trivially same as $l_i = 1$
L-constraint	$O(n)$ time, $O(n)$ space [H]	$O(n)$ time, $O(1)$ space [*]
U-constraint or L- & U-constraints	$O(n)$ time, $O(n)$ space [LJC]	$O(n)$ time, $O(n)$ space [*]

[K]: Kadane as reported by Bentley 1984/2002

[H]: Huang 1994

[LJC]: Lin-Jiang-Chao 2002

[*]: This paper

This paper improves some time and space bounds as well as achieving results applicable to more general models. E.g., whatever appears in right-hand column is also applicable to left.

Finding Densest Segments — Results

	atomic $l_i = 1$	atomic $l_i \geq 1$	$l_i \geq 0$
L-constr.	$O(n \lg L)$ time and space [LJC]	$O(n)$ time and space [GKL]	$O(n)$ time & space [*]
L- & U-constr.	$O(n)$ time and space [GKL]	$O(n \lg(U - L + 1))$ time and space [GKL]	$O(n)$ time & space [*]

[LJC]: Lin-Jiang-Chao 2002

[GKL]: Goldwasser-Kao-Lu 2002

[*]: This paper

Anything without an L-constraint is trivially $O(n)$ time and $O(1)$ space. $n \lg(U-L+1)$ is really $n + \ldots$. Again, results appearing in one column are applicable to columns to left.

Some Algorithmic Details

- Non-uniform lengths
- Constant space for a heaviest segment with no U-constraint
- Non-atomic runs

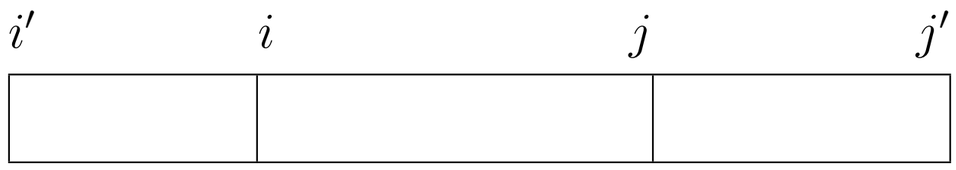
Non-Uniform Lengths

Algs. of Kadane, Lin et. al., and Goldwasser et. al. can all be cast in linear sweep form:

- Move right endpoint of potential optimal segment left to right across sequence.
- At each step, determine best choice of left endpoint ("good partner")
- Good partner can be restricted to move left to right.

Good Partner Restriction

If best left endpoint requires backing up, then current right endpoint is not right endpoint of an optimal segment. E.g., for density:



$$density(j, j') < density(i', i) \leq density(i, j)$$

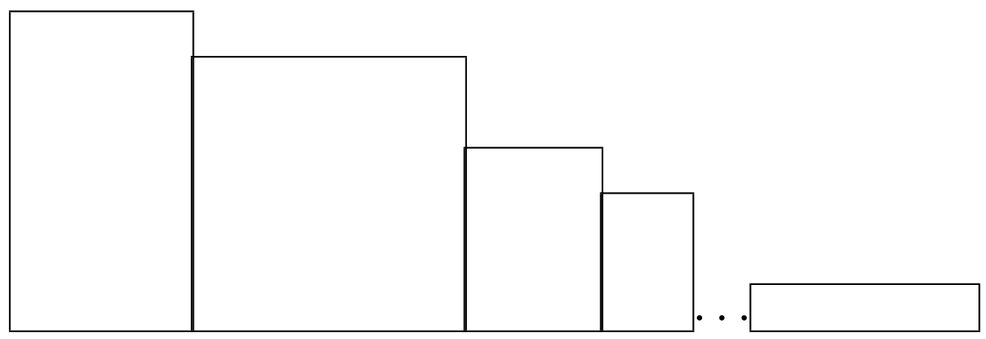
implies

$$density(i', j') < density(i, j)$$

Avoiding Backup

- For problems with one constraint type, some linear time precomputation can be utilized to allow constant time determination of next left endpoint position that is an improvement.
- For density, basic idea is finding decreasing right-skew partitions.

Decreasing Right-Skew Partition



$density(A_1) > density(A_2) > \cdots > density(A_m)$ and each of A_1, A_2, $\ldots A_m$ is right skew:

$$density(i, j) \leq density(j + 1, k) \quad \forall i \leq j < k$$

Non-Uniform Wrap-Up

- One remaining detail is to maintain length constraint on good partner. Can be managed by adding and subtracting lengths during left-to-right scan.
- For density with L- and U-constraints, begin by dividing sequence into blocks of length approx. $U - L$ so finding good partner breaks down into:
  1. Find in a specific block w. no explicit U-constraint.
  2. Find in next farther block w. no explicit L-constraint.

Constant-Space Heaviest w. L-Constraint

Generalize Kadane. Uniform here:

```
1     maxsofar ← maxendinghere ← 0
2     for i = 1 to n do
3         maxendinghere ← max{maxendinghere + w_i, 0}
4         maxsofar ← max{maxsofar, maxendinghere}
5     endfor
```

⇒

```
1     maxsofar ← maxendinghere ← Lhere ← Σ_{i=1}^{L} w_i
2     for i = L + 1 to n do
3         Lhere ← Lhere + w_i − w_{i−L}
4         maxendinghere ← max{maxendinghere + w_i, Lhere}
5         maxsofar ← max{maxsofar, maxendinghere}
6     endfor
```

Non-Atomic Runs

- Need only consider segments with a partial run on at most one end.
- Need only consider a segment containing a partial run if its length is L or U.

Conclusion

- Extended all results to non-uniform lengths with non-atomic runs. Enables working with run-length encoded sequences.
- Improved time and space bounds for finding a densest segment with L- and U-constraint to $O(n)$.
- Improved space bound for finding a heaviest segment with an L-constraint to $O(1)$.
- Some other results in paper, e.g., same bounds for finding *all* optimal segments of minimal length.