Instance-Optimal Computational Geometry Made Easier and Sensitive to Sortedness

David Eppstein   Michael T. Goodrich   Abraham M. Illickan   Claire A. To

Department of Computer Science at University of California, Irvine

Introduction

Instance-optimal algorithms achieve performance as good as any correct algorithm on every input instance up to a constant factor with respect to a given measure.

Real-world data often has underlying patterns or structures that are not captured by traditional worst-case analysis. These hidden properties can be leveraged to improve the efficiency of algorithms.

We study how to exploit input *sortedness* in geometric problems.

Motivation & Problem

Previous work focused on *output size* and *spatial distribution* for instance-optimality. However, the role of input sortedness remains unexplored in computational geometry.

- **Sortedness:** Measures how close the input is to being sorted, such as through *inversions*, *removals*, and *runs*.
- **Shannon sequential entropy:** Measures the degree of order.
- **Shannon structural entropy:** Measures the placement and spread.

Our goal: Define a complexity measure that captures sortedness and structure to design and analyze algorithms that adapt to it.

New Complexity Measure

Shannon range-partition entropy combines Shannon sequential and structural entropy and subsumes both.

Given a set, S, of n points, a partition, Π, of the set into disjoint subsets is *respectful* if:

1. **Local property:** Fulfills properties within a subset.
2. **Global compatibility:** Fulfills dependencies between subsets.

The *entropy*, $\mathcal{H}(\Pi)$, of a partition, $\Pi = \{(S_1, R_1), \ldots, (S_t, R_t)\}$, is

$$\mathcal{H}(\Pi) = -\sum_{i=1}^{t} \left(\frac{|S_i|}{n}\right) \log \left(\frac{|S_i|}{n}\right).$$

The *range-partition entropy*, $\mathcal{H}(S)$, of S is the minimum $\mathcal{H}(\Pi)$ over all respectful partitions.

References

[1] Peyman Afshani, Jérémy Barbay, and Timothy M. Chan. Instance-optimal geometric algorithms. *Journal of the ACM*, 64(1):A3:1–A3:38, March 2017.

[2] Nicolas Auger, Vincent Jugé, Cyril Nicaud, and Carine Pivoteau. On the worst-case complexity of TimSort, 2019. Previously announced at ESA 2018.

[3] David G. Kirkpatrick and Raimund Seidel. Output-size sensitive algorithms for finding maximal vectors. In *1st ACM Symposium on Computational Geometry (SoCG)*, page 89–96, 1985.

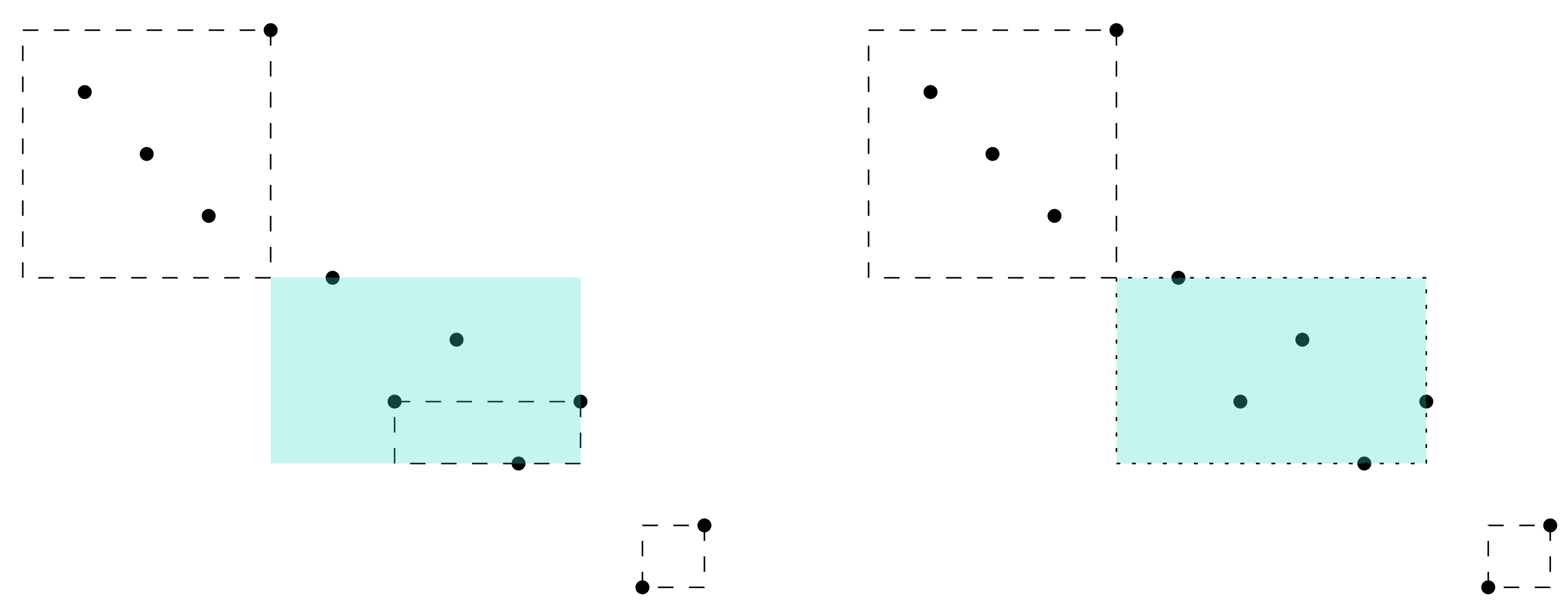
[4] David G. Kirkpatrick and Raimund Seidel. The ultimate planar convex hull algorithm? *SIAM Journal on Computing*, 15(1):287–299, 1986.

2D Maxima Set

Problem: Find the subset of points that are not dominated by any other point (no other point has both greater x- and y-coordinates).

Algorithm: Divide-and-conquer algorithm [1, 3]. Before recursively solving a subset, check if the points are sorted; if so, compute the maxima set in linear time.

Analysis: Leverages structure and sortedness. Runs in $O(n(1 + \mathcal{H}(S)))$ time, where $\mathcal{H}(S)$ is the range-partition entropy of S.



(a) (b)

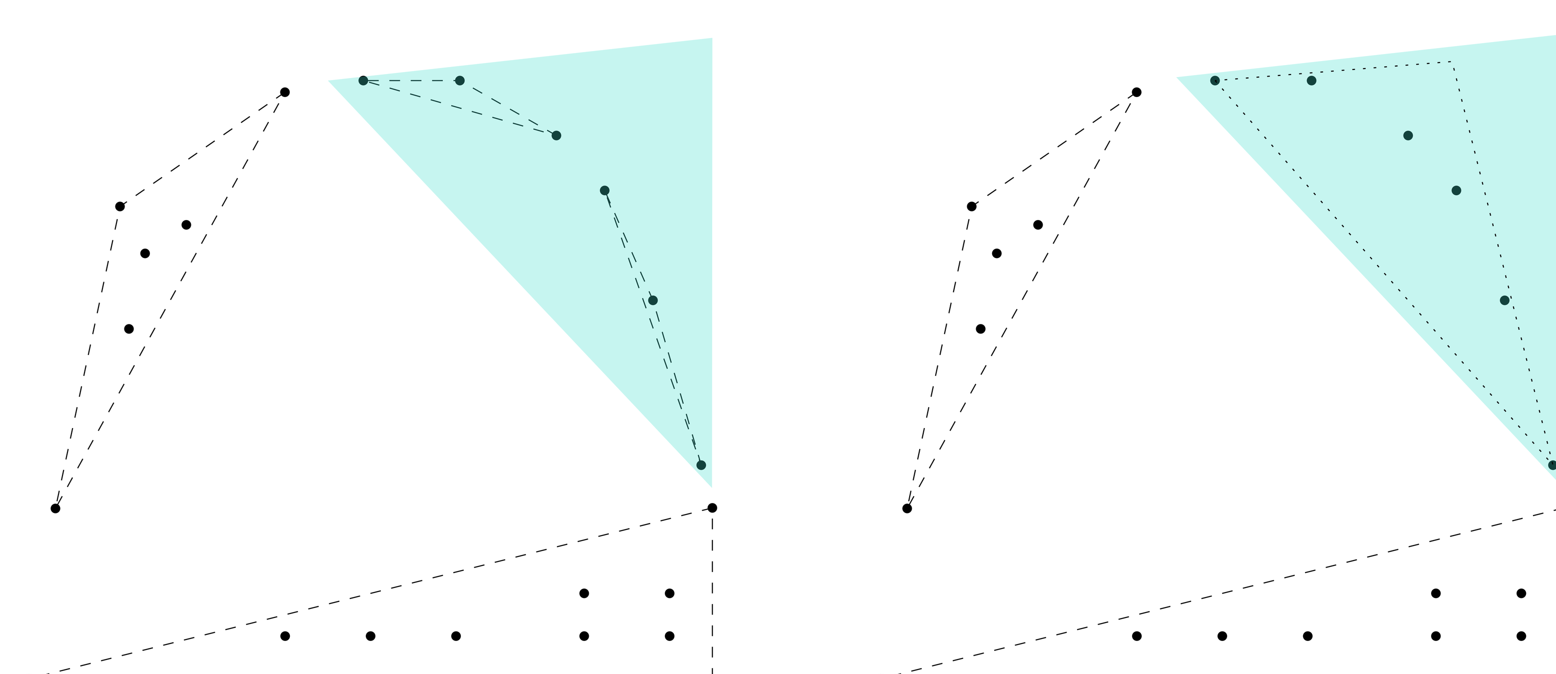
Figure 1. Respectful partitions for maxima set. The points in the blue shaded rectangle are sorted among themselves. (a) A respectful partition without using sorted subsets. (b) A respectful partition using both types of sets, leveraging sortedness.

2D Convex Hull

Problem: Find the smallest convex polygon enclosing all points.

Algorithm: Divide-and-conquer algorithm [1, 4]. Before recursively solving a subset, check if the points are sorted; if so, compute the convex hull in linear time.

Analysis: Leverages structure and sortedness. Runs in $O(n(1 + \mathcal{H}(S)))$ time, where $\mathcal{H}(S)$ is the range-partition entropy of S.



(a) (b)

Figure 2. Respectful partitions for convex hull. The points in the blue shaded triangle are sorted among themselves. (a) A respectful partition without using sorted subsets. (b) A respectful partition using both types of sets, leveraging sortedness.

3D Convex Hull

Problem: Find the smallest convex polyhedron enclosing all points.

Algorithm: Iteratively partition and prune [1]. Partitioning via eight-sectioning in expected linear time with random sampling and combinatorial partitioning. Recursive partitioning runs in $O(n \log n)$ time.

Analysis: Leverages structure with improved partitioning. Runs in $O(n(1+\mathcal{H}(S)))$ time, where $\mathcal{H}(S)$ is the range-partition entropy of S.

Lower Envelope

Problem: Find the vertical point-wise minimum of a set of disjoint monotone line segments.

Algorithm: Stack-based mergesort algorithm with redefined weights.

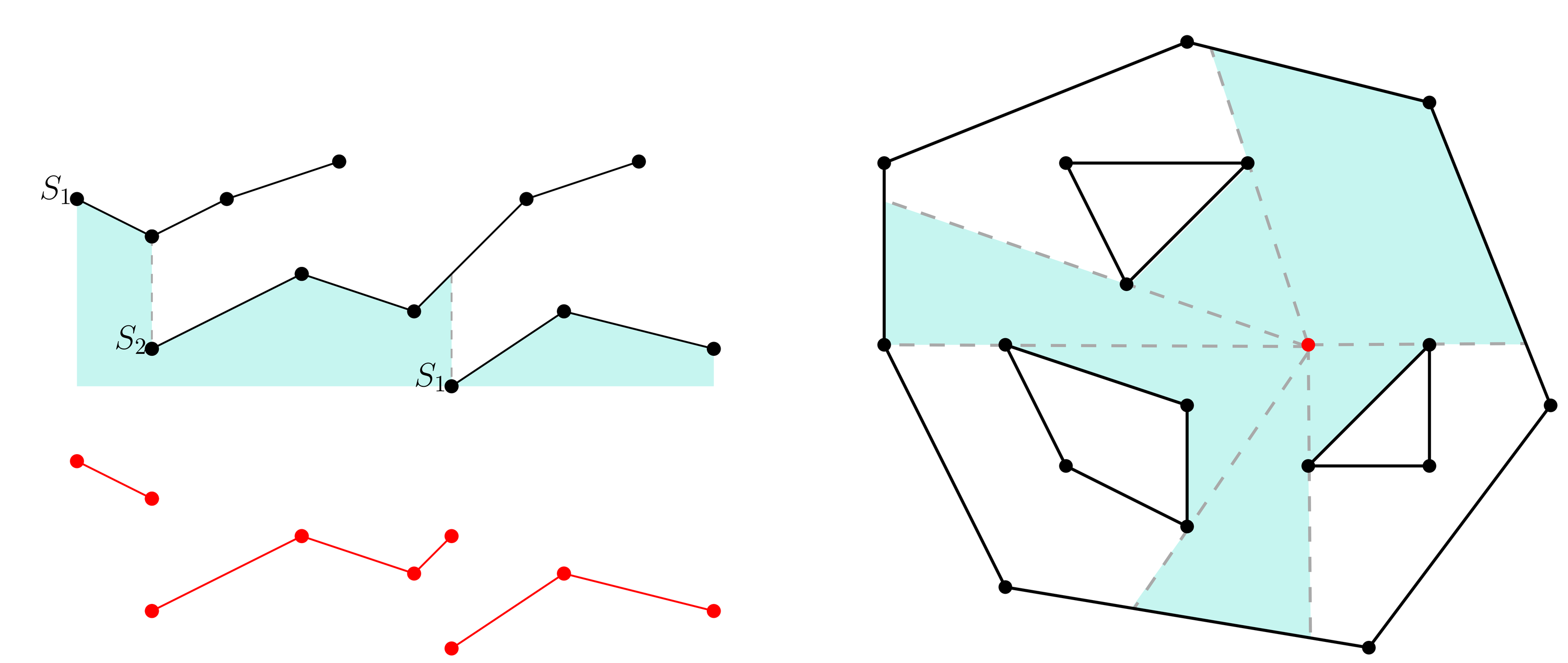
Analysis: Leverages monotonic runs similar to TimSort [2]. Runs in $O(n(1+\mathcal{H}(S)))$ time, where $\mathcal{H}(S)$ is the range-partition entropy of S.

Visibility Polygon

Problem: Find the radial point-wise minimum, the visible region, from a point inside a convex polygon.

Algorithm: Stack-based mergesort algorithm with redefined weights.

Analysis: Reduction from lower envelope. Runs in $O(n(1 + \mathcal{H}(S)))$ time, where $\mathcal{H}(S)$ is the range-partition entropy of S.



(a) (b)

Figure 3. (a) Merging two sets of disjoint monotone chains. S_1 has two sequences, S_2 has one. (b) The visibility polygon of a point among disjoint convex chains.

Conclusion

Sortedness is a powerful but overlooked property in instance-optimal analysis. By leveraging sortedness, we can design algorithms that are more efficient than traditional worst-case approaches. We demonstrated how algorithms for classical geometric problems can benefit from recognizing and exploiting input sortedness.

claire.to@uci.edu