Inference Maximizing Point Configurations for Parsimonious Algorithms

ADG 2025

Shivam Sharma and John Keyser

Texas A&M University

<2025-08-01 Fri>

Outline

Section 1

Introduction

The 2 approaches for Geometry

Synthetic

- Developed by Euclid, this approach uses axioms and theorems to prove "equality" of entities (e.g. congruence and similarity)
- There are no "coordinates" in this, no use of numbers

Analytic

- Developed in 17th century by Descartes, it brings "coordinates", and enabled algebraization of geometry, which revolutionized Geometry significantly

How Computational Geometry (CG) is done

- CG is conventionally done in Analytic fashion (i.e. using coordinates and numbers)
- CG involves both numerical computation and discrete decision making (combinatorial computation)
- A single wrong discrete decision (done using floating point computation) completely changes the direction of the algorithm
  - This causes Robustness issues in CG

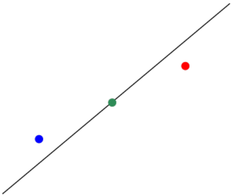


Figure: A Fundamental CG Primitive

A curious question: A Hybrid Approach?

"*Can we combine the Analytic approach with a Synthetic approach for Computational Geometry?*"

Key Motivation for Hybrid Approach: Robust Computational Geometry

- Computational Geometry fails primarily owing to the failures in establishing "equality"
- Synthetic Geometry deals exclusively with notions of "equality" (such as congruence and similarity). Hence, it is fitting to utilize Synthetic Geometry.

Prior Attempts at Hybrid or Synthetic Computational Geometry

- Bokowski, J., & Sturmfels, B. (1989). Computational synthetic geometry.
- Knuth, D. E. (1992). Axioms and hulls.
  - Knuth coined the term "Parsimonious Algorithms" for such hybrid algorithms: *they never numerically compute anything that can be "deduced" from prior computations*
- Recent works from Homotopy Type Theory Research, e.g. Synthetic Differential and Algebraic Geometry

Our work focuses on the "efficacy" analysis of Knuth's parsimonious algorithms.

Section 2

Work Overview

Parsimonous Algorithm Definition

Parsimonious Algorithm

We say that an algorithm is *parsimonious* if it never makes a test for which the outcome could have been inferred from the results of previous tests, with respect to a given set of axioms.

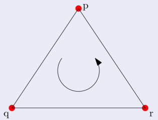
The Geometric Problem: Order-type Calculation

Our Geometric Problem

- Given a set of n points $P = \{p_1, p_2, \ldots, p_n\}$ in 2D, compute its order type parsimoniously

CounterClockwise (CC) Relation (or Orientation)

- For 3-point Orientation Test
- pqr is True, if $p \rightarrow q \rightarrow r$ is Counter-Clockwise
- Can be used for Convex Hulls and many subsequent computational geometry algorithms



Order Type of Point Set

For a 2D point set, the order type is the mapping from all its ordered triples to their orientations

CC Relation Axioms

1. Cyclic Symmetry: $pqr \Rightarrow qrp$
2. Anti-symmetry: $pqr \Rightarrow \neg prq$
3. Non-degeneracy: $pqr \vee prq$
4. Interiority: $tqr \wedge ptr \wedge pqt \Rightarrow pqr$
5. Transitivity: $tsp \wedge tsq \wedge tsr \wedge tpq \wedge tqr \Rightarrow tpr$

The first three can be captured within a data structure. We wish to do deductions using (4) and (5). For this work, we focus on (4).

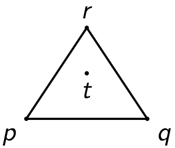


Figure: Axiom 4 of CC Relation

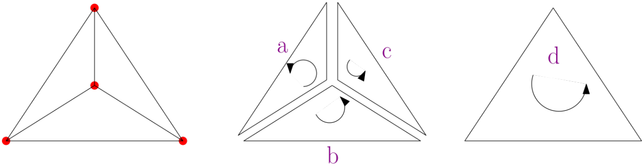
A Sample Parsimonious Algorithm

Algorithm 1: Order Type Parsimonious Computation using Axiom-4

Input: Point set $P = \{p_1, p_2, \dots, p_n\}$
Output: Set of CC relations

$\mathit{CCRelations} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** $n - 2$ **do**
    for $j \leftarrow i + 1$ **to** $n - 1$ **do**
        for $k \leftarrow j + 1$ **to** n **do**
            $\mathit{deduced} \leftarrow \texttt{DeduceUsingAxiom4}(p_i, p_j, p_k, \mathit{CCRelations})$;
            if $\mathit{deduced} \neq \varnothing$ **then**
                $\mathit{CCRelations} \leftarrow \mathit{CCRelations} \cup \{\mathit{deduced}\}$;
            else
                $\mathit{computed} \leftarrow \texttt{ComputeCCRelation}(p_i, p_j, p_k)$;
                $\mathit{CCRelations} \leftarrow \mathit{CCRelations} \cup \{\mathit{computed}\}$;
return $\mathit{CCRelations}$;

A Sample Parsimonious Algorithm in action



- ✓ Parsimonious: $a(N) - b(N) - c(N) - d(D)$
- ✓ Parsimonious: $d(N) - a(N) - b(N) - c(N)$
- × Not Parsimonious: $a(N) - b(N) - c(N) - d(N)$

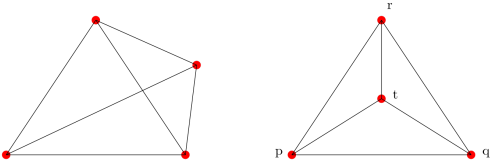
Overview of our Work

- We do an "efficacy" analysis of Knuth's Parsimonious algorithms. That is, we ask: *how many maximum primitives can be "inferred" for a given geometric problem (as compared to numerically computed)*
- The geometric problem we choose for this analysis is: *order-type calculation of a random 2D point-set*

Section 3

The Efficacy Analysis

Efficacy Analysis Definition



Minimum Inference Case (= 0)

Maximum Inference Case (= 1)

Possible Sequences = $4! = 24$
$3!$ (= 6) yield max inferences
One such sequence:
tpq - tqr - trp - pqr
Δ^0-Δ^0-Δ^0-Δ^1

Figure: The only 2 configurations for n=4 case

Goal:

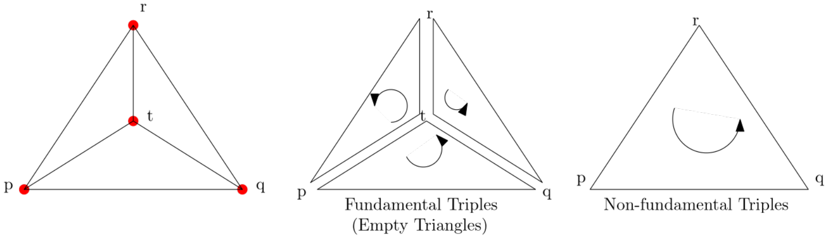
Given a parsimonious algorithm, using axiom-4 for inferences, operating on n-point configurations, we have 2 questions:

1. Which point configurations give rise to the maximum inferences?
2. Which computation sequences achieve that maximum number?

Minimal Configurations are Trivial

When all points on Convex Hull, number of inferences will be 0.

Fundamental vs Non-fundamental Triples



Fundamental Triple

A fundamental triple is a triple with no points in it. They are represented as Δ_0. They can never be deduced and always have to be numerically computed.

Non-fundamental Triple

A non-fundamental triple is a triple with at least one point in it. They are represented as Δ_i where $i \geq 1$. They can always be deduced (we have a theorem proving this).

Fundamental Triple $\equiv$ Empty Triangle

- For inference using axiom-4, fundamental triples happen to be the same as empty triangles. For inferences using axiom-5 (and other theorems), fundamental triple set may not be equivalent to the empty triangle set.
- Hence, our current problem is the same as the *"Minimum Empty Triangle Problem"*, an Erdős-type problem in Discrete Geometry research

A Glance at Problem Complexity

- Total Number of All Triangles(m): $\binom{n}{3}$
- Number of Fundamental Triangles: No known formula
- Number of Non-fundamental Triangles: $\binom{n}{3}$ − Number of Fundamental Triangles
- Number of possible sequences for deduction: $m!$, where $m =$ Number of all Triangles

For 6 points:

- there are $C_3^6 = 20$ triples
- number of possible sequences of triple computations: $20! = 2432902008176640000$

No. of Points	3	4	5	6	7	8	9	10
No. of Order Types	1	2	3	16	135	3,315	158,817	14,309,547

Table: Order Type Complexity

The Constructive Approaches

$$\#\text{MaxDeductions} = \binom{n}{3} - \#\text{FundamentalTriangles} \tag{1}$$

We have to maximize (1). But there is no straight formula for $\#FundamentalTriangles$, and therefore, neither for (1). So, we suggest 2 constructive approaches to figure out the maximum deduction case:

1. Forward Construction
2. Backward Construction

Section 4

Efficacy Analysis 1: Extremal Configurations

Forward Construction

- We start with $k < n$ points in plane in non-degenerate position
- We add one point in each of the different regions separately, hence generating new $(k + 1)$-point configurations
- We weed out isomorphic configurations, hence unique configurations remain
- Keep doing this till we reach n

Key Idea

Doing such constructions could lead to insights about which “type” of constructions lead to maximum deductions. We start with k=3 (i.e. an empty triangle case) and add points one-by-one.

Forward Construction Examples

We did by hand for 4-point, 5-point, and 6-point cases.

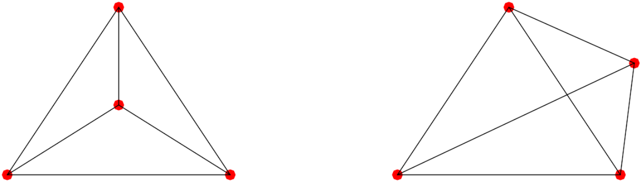


Figure: n=4 results in 2 unique order types

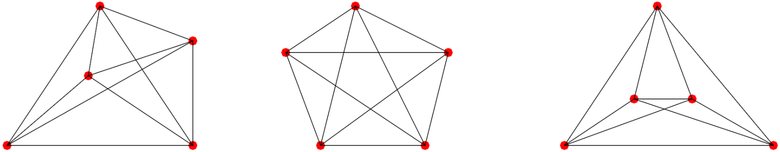
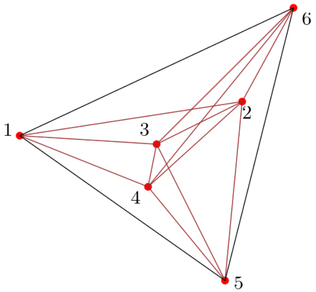


Figure: n=5 results in 3 unique order types

Forward Construction 6-point Example



Inferred:

1 - 3 - 5
1 - 2 - 4
1 - 2 - 5
1 - 6 - 5
1 - 6 - 4
6 - 4 - 5
6 - 3 - 5

Figure: The maximal case for n=6

Backward Construction

- Assume that we have been given the convex hull of a maximal point configuration of size n.
- Connect all points with lines, giving rise to different regions
- We add one point in one of these regions and see which region maximizes the inferrable triples (i.e. minimizes empty triangles)

Backward Construction Example

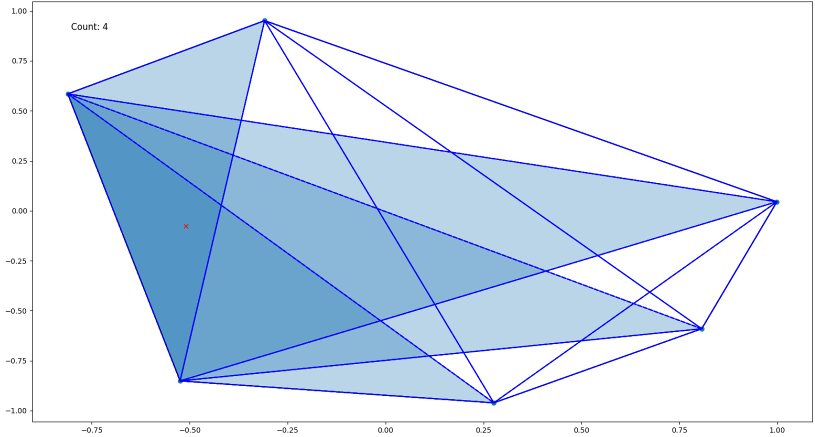


Figure: Backward Construction for n=6

(contd..) Backward Construction Example

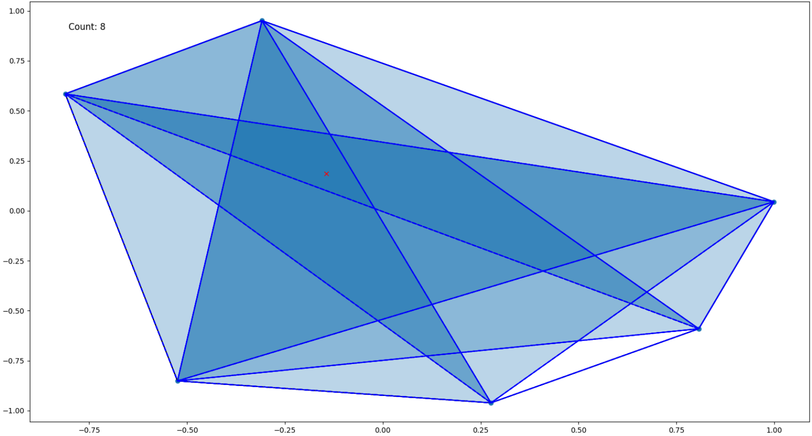
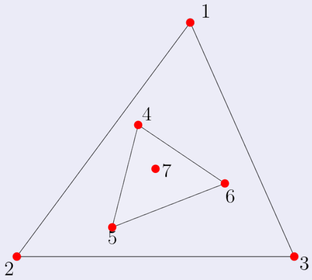


Figure: Backward Construction for n=6

Candidates for Inference Maximizing Configuration

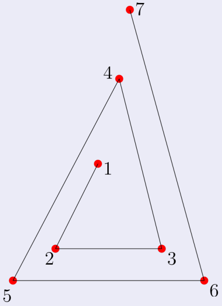
Triangle-within-Triangle configuration

Using the ConvexHull Heuristic



Spiral Point Configuration

We do a forward construction and start placing points in an outward spiral



Section 5

Efficacy Analysis 2: Inference Maximizing Sequences

Problem Complexity

- Δ^0: Fundamental Triples
- Δ^i $(i > 0)$: Non-fundamental Triples with "i" interior points
- Trivial Maximal Sequences: $\Delta^0\Delta^0 \ldots \Delta^i\Delta^i \ldots \Delta^j\Delta^j \ldots$, such that $j > i$
  - where you first compute all of Δ^n before proceeding to Δ^{n+1}

Possible Computation Sequences

$$\boxed{T(n) = O\big((n^3)!\big)}$$

Maximal Sequences' Properties

Theorem-1 [Existence]:

An inference-only path always exists for all non-fundamental triples.

Theorem-2 [Non-Trivial Sequences]

Non-trivial maximal sequences exist.

Theorem-1 Proof Sketch

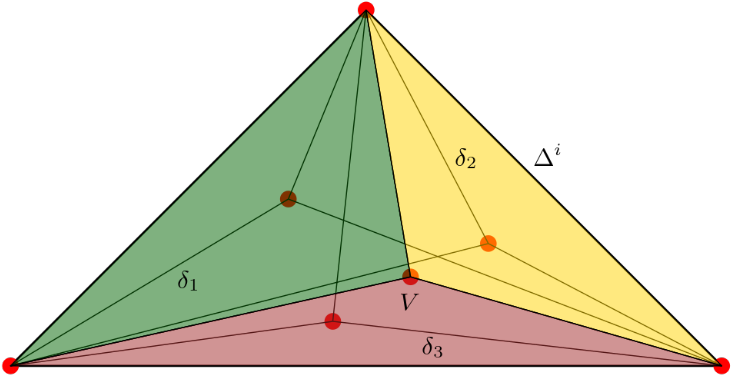


Figure: Inductive Step of the Proof

Section 6

Summary

Immediate Next Steps

1. Automating Constructions using Order-type Isomorphism
2. Proving the Greedy Property for Constructions
3. Counting Maximal Sequences
   - Using hypergraph approach

Future Work and Extensions

1. Utilizing other axioms and theorems for inference
   - Multi-axiom Inference
2. Connecting to Exact and Robust Computation
3. Neural-network Assisted Automated Constructions
4. Getting tighter bounds "*minimum empty triangle problem*"

Questions

Questions?
Comments?