Inference Maximizing Point Configurations for Parsimonious Algorithms

Shivam Sharma[1][0000−0003−1480−7684] and John Keyser[1][0000−0002−4829−9975]

Texas A&M University, College Station, TX 77840, USA

Abstract. We present an exploration of inferring orientations for point configurations. We can compute the orientation of every triple of points by a combination of direct calculation and inference from prior computations. This "parsimonious" approach was suggested by Knuth in 1992 [6], and aims to minimize calculation by maximizing inference. We wish to investigate the efficacy of this approach by investigating the minimum and maximum number of inferences that are possible for a point set configuration. To find the configurations which yield maximum inferences, there is no direct formula and hence two constructive approaches are suggested. A basic analysis of sequences that achieve those maximum inferences is also presented and some properties have been proved regarding their existence.

Keywords: Combinatorial Geometry · Order Types · Robust Geometric Computation.

1 Introduction

Knuth [6] suggested a hybrid approach for *Computational Geometry* in which computation for geometric predicates starts numerically but switches to inference (based on prior computed predicates) wherever possible. He termed this approach "parsimonious", since in this approach, you never numerically compute any predicate that can be inferred - i.e. you always remain "parsimonious". Such parsimonious approaches have interesting implications for robustness in geometric computation, where program failures often occur due to conflicting predicate evaluations; a parsimonious approach could potentially ensure consistency (and thus robustness) even if not guaranteeing accuracy. Further, since inference would retain the "exactness" of a computation, if the numerical calculations or prior knowledge were guaranteed exact, the overall appraoch would be exact.

One of the geometric predicates Knuth considered, and the one we focus on in this paper, was the "orientation" predicate (called "CounterClockwise" (CC) predicate) for an ordered triple of points. Five axioms were laid down for this predicate and several theorems were derived. These axioms and theorems could then be used to infer the orientation for certain triples from prior computed orientations of other triples. Our focus here is on inference from the first four of these axioms.

However, this inference is not always possible for all triples for several reasons. One, there is a dependency among triples – to infer some triples, you have to pre-compute some other triples (either numerically or inferentially). Hence, the order of computation matters. Second, in this dependency graph, some triples are "fundamental" and can only be computed numerically. Third, different point configurations will have different numbers of these "fundamental" triples. This third point is the focus of our work. We wish to investigate which configurations give rise to minimum and maximum number of inferences. We collectively call these the "extremal configurations". Getting these minimum and maximum numbers would give us insight into the best-case and worst-case performance of the "parsimonious" algorithms suggested by Knuth [6].

Finding the inference minimizing configuration was trivial, and the answer is a point configuration with all points on the convex hull; this will make the number of possible inferences to be 0 (zero). To find the inference maximizing configuration, we developed two constructive approaches that we label Forward and Backward Construction. The purpose of these constructive approaches is to maximize inference for fixed numbers of points, in turn revealing patterns which lead to inference maximizing configurations for any number of points.

The paper is organized as follows. In Section 2, we first discuss the required background which includes the axiomatic framework of Knuth [6] and the concept of order types and point configurations [7,4,3]. In Section 3, we then lay out our problem of finding extremal configurations, and discuss in detail our approach to finding them. In Section 4, we discuss the results for maximal configuration. In Section 5, we conclude with a discussion of open questions for future work.

The key accomplishments we present here are as follows:

- We have defined the problem of finding extremal configurations (which is not done in prior literature)
- We have a conclusive (though trivial) answer for minimal configuration
- We propose two constructive approaches to find maximal configurations
- We have proved that there always exists an inference sequence such that all non-fundamental triples will be inferred
- We have proved the existence of non-trivial inference sequences
- We have done basic counting for non-trivial inference sequences, and demonstrated that it's a larger problem involving dependency graph among triples

2 Background

2.1 Parsimonious Computational Geometry

In this section, we briefly describe an axiomatic computational geometry approach that was suggested by Knuth in [6]. He focused on two geometric predicates for points in R^2: (1) the CounterClockwise predicate on ordered triples (called CC relation), and (2) the InCircle predicate on ordered quadruples (called CCC relation). The first predicate (CC relation) was used to develop a Convex

Hull algorithm that solely used the CC relation. The second predicate (CCC relation) was used to develop a Delaunay Triangulation algorithm that solely used the CCC relation. Both these algorithms could act "parsimoniously": that is, they would numerically compute only as many predicates as could not be inferred, and would always infer predicates if an inference was possible.

Definition 1 (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.*

Definition 2 (CC Relation/Orientation). *We define the* ***CounterClockwise (CC) Relation*** *(also called* ***Orientation****) of an ordered triple of points pqr to be one of the three numbers $+1$, $+1$, or 0. It is $+1$ if the circle through the points p, q, r is traversed counterclockwise when we encounter the points in the cyclic order $p \to q \to r \to p$. It is -1 if it is traversed clockwise, and it is 0 if the three points are collinear.*

In our case, we will consider all the points to be in general position (i.e. no three points collinear), hence orientation 0 will never occur.

Axioms for CC Relations Knuth [6] had laid down 5 axioms for the CC system:

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: $tsp \wedge tsq \wedge tsr \wedge tpq \wedge tqr \Rightarrow tpr$

A demonstration for axiom-4 can be seen in Figure 1. The orientation of 3 interior triples can be used to infer the orientation of the outer triple pqr.

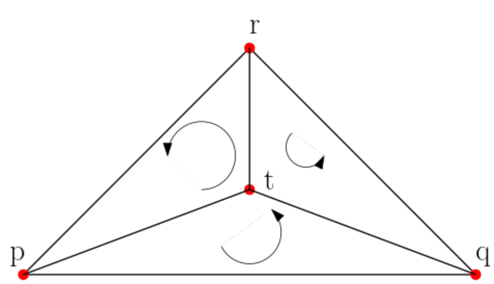


Fig. 1. Demonstrating Axiom 4 ($tqr \wedge ptr \wedge pqt \Rightarrow pqr$)

The focus of our work is on the first 4 axioms, especially on the axiom-4. Below, when we use the term "infer" we will generally be referring to using axiom-4 to determine a CC relation given 3 other CC relations. In our implementation of a parsimonious CC algorithm, the first 3 axioms were somewhat trivial and embedded within the data structure itself (hence we won't be discussing them), and we used only axiom-4 for deductions. We emphasize that our goal here is not to understand how much can possibly be inferred (though other axioms or variations on these) regarding CC relations, but rather to explore, given a strict limitation in what we can apply (only direct application of axiom-4), what the limits of inference are.

2.2 Order Types of Point Configurations

We will use the definitions from [3] to define Order-type, Order-equivalency, and Configuration.

Definition 3. *The **order type** of a set of points in the Euclidean plane is the mapping from ordered triples of points to their orientations.*

Definition 4. *Two finite sets of points S and T are considered **order-equivalent (or isomorphic)** if we can map the points of S one-to-one onto the points of T in a way that preserves the orientation of each triple of points.*

Definition 5. *We define the **Configuration** of a point set S to be the isomorphism class of all finite sets of points that are order-equivalent to S. In our work, we will use "configuration" and "order type" interchangeably, since they are often used interchangeably in the literature as well.*

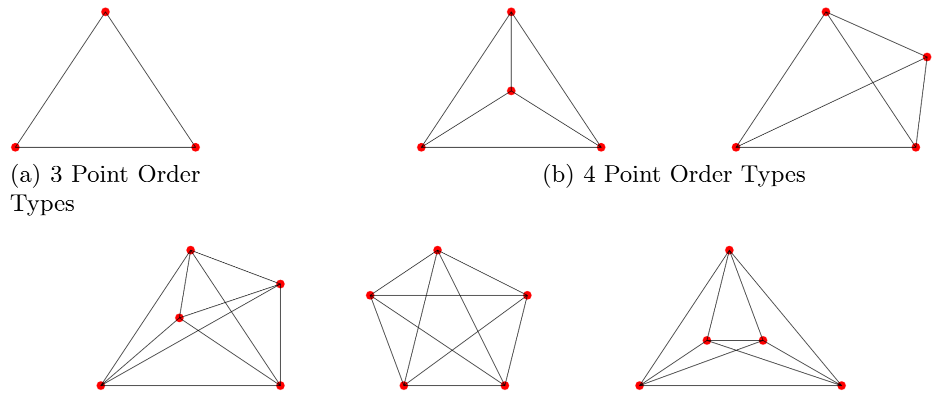
Figure 2 shows the order types for 3, 4, and 5 points respectively. The number of order-types (or configurations) of n points grow very quickly, as $c^{n \log n}$ for a constant c $>$ 1. Order Types have been computed [1] using computer search for up to 11 points, and the following results have been found:

Number of Points	3	4	5	6	7
Number of Order Types	1	2	3	16	135

Table 1. Order Types

2.3 Fundamental vs Non-fundamental Triples

As mentioned in the previous section, we focus on parsimonious CC inferences using Axiom-4. In the simplest example of this axiom at work (in Figure-1), we see that this axiom essentially implies that triples which have no points in them will necessarily have to be numerically computed, and the triples that have a point in them can always be inferred (we prove in a later section that this is guaranteed if we do our computation in the right sequence). This gives us a clear demarcation of the two types of triples:



(a) 3 Point Order Types

(b) 4 Point Order Types

(c) 5 Point Order Types

Fig. 2. Order Types

1. **Fundamental Triples**: those triples which can not be inferred and have to be numerically computed. For axiom-4, these triples are the ones with no points in them. That is, they are *"Empty Triangles"*.
2. **Non-Fundamental Triples**: a non-fundamental triple is one that can always be inferred. For axiom-4, these triples are the ones with at least one point in them.

3 Efficacy Analysis of Parsimonious Algorithm

Knuth's complexity analysis for parsimonious algorithms for the CC-system proved that finding a parsimonious algorithm for a particular situation is NP-Complete [6]. We instead wish to ask questions related to the "efficacy" of these parsimonious algorithms: that is, how "effective" they would be in practice. So, in a given random set of points, if we implement a parsimonious algorithm (e.g. *Convex Hull* or *Delaunay Triangulation*), how many computation steps can be done parsimoniously? To answer this question for a specific algorithmic problem (say Convex Hull) would be challenging, so we ask this question for the simplest geometric problem for planar point sets: "for computing the orientations of all triples of a planar point set, how effective would the parsimonious approach be?" We define "efficacy" in this case through 2 sub-questions:

1. How many maximum deductions can our parsimonious approach theoretically make? Are there certain **extremal configurations** that maximize or minimize the number of possible deductions?
2. What is the average number of deductions we can make, probabilistically speaking? This problem entails analyzing the different **inference sequences** and calculating their probabilities.

We now address both these problems in the following 2 sections.

3.1 Problem-1: Extremal Configurations

Minimal Configuration A minimal configuration would minimize the number of inferences that could be made, and so one question is what the smallest number of inferences could be. The answer to this is trivial: 0. If we have all the n points on the convex hull, all the triples formed are empty triangles, and hence all of those triples will need to be computed. Thus we will have 0 inferences.

Maximal Configuration Likewise, we would like to know the maximum number of inferences possible. Unfortunately, there is no direct answer to this question. The problem of maximizing the number of non-fundamental triples, N_{NF}, is the same as minimizing the number of fundamental triples (i.e. empty triangles), N_F. This problem of "*Minimum Empty Triangles*" has been well studied in the Discrete Geometry and Graph Drawing communities [5,2], and upper and lower bounds for this number have been found. The upper bound for minimum empty triangles has been proven to be at most linear in n. However, there is still no known formula or procedure for finding the minimum number of empty triangles.

To tackle this problem, we have taken an approach of investigating through constructions. We construct point configurations which *may* maximize N_{NF} (and hence, minimize empty triangles). We propose 2 types of constructions for this – Forward Construction and Backward Construction. These constructions can reveal patterns to us about what type of configurations yield the maximal N_{NF} values. This may, in turn, also provide answers to the classic Discrete Geometry problem of empty triangles.

In the **Forward Construction**, we start with the 3-point non-degenerate case (which is a triangle), then place new points one by one in different regions, and count the maximum possible inferences in each of the new configurations that arises after new point placement. We "greedily" choose the configuration that yields maximum inferences, and then add new points in its regions to generate further new configurations, and repeat the process. We did this manually for up to 6 points, however possible configurations explode in number quickly. This process can be automated in future using the concept of "order type isomorphism", but this is beyond the scope of our current work.

Note that forward construction is guaranteed to give you the optimal answer for that number n (up to which have done the construction). This is so because forward construction is an exhaustive search.

In the **Backward Construction**, we assume that we already have an inference maximizing configuration, and we know its Convex Hull (but not the full configuration). We now construct the remaining part of the configuration starting from its Convex Hull. To do this construction, we place points greedily in regions inside the convex hull such that the number of inferences are maximized after a point placement. We hypothesize that this greedy placement will lead to overall maximal configuration.

3.2 Problem-2: Inference Maximizing Sequences

For a given point configuration, the sequence in which we compute our orientations matters. For example, in Figure 1, if pqr is computed first, then we'd have to necessarily calculate it and can not infer it. Since remaining triples are fundamental, they also can not be inferred (by axiom-4 alone). This leads to zero inferences. However, if pqr was computed last, it could have been inferred and we'd have a total of 1 inference (which is the maximum for this case). This calls for a study of **computation sequences** - the order in which geometric entities are computed dictates if we realize the theoretical maximum number of inferences for that point configuration.

Now, let us first prove the existence of inference maximizing sequences (or maximal sequences), i.e. sequences which achieve the theoretical maximum inferences for a given point configuration. Alternatively, we can say that if an inference path exists for all non-fundamental triples and they are always inferred, then we achieve maximum inferences. We prove this in Theorem-1 below.

Theorem 1 (Maximal Sequences Exist). *An inference-only path always exists for all non-fundamental triples.*

Proof. We can prove this using induction.

Let us denote our triples as follows:

- Δ^0: a fundamental triple
- Δ^n: a non-fundamental triple with n points enclosed within it

Now, first we calculate all fundamental triples. Then, consider Δ^1. Each Δ^1 can be decomposed into 3 Δ^0's, all of which have already been computed. Hence, each Δ^1 can be inferred. This is our base case of the induction proof ($i = 1$).

Now, let's prove the inductive step. Say, each of $\Delta^{j<i}$ has already been inferred (our induction hypothesis). We have to prove that each Δ^i can then also be inferred. Now, observe that when we choose any vertex within Δ^i, it will create 3 triples that share a vertex, as shown in Figure 3. Here, we choose three triples δ_1, δ_2 and δ_3 (out of many other triples that exist inside Δ^i) sharing vertex V. Since Δ^i contains i points and one point (V) is used up as the common vertex, each of these triples must have no more than $i-1$ points within it. However, all Δ^{i-1} and smaller triples have already been computed at this point (our induction hypothesis), thus we can also infer Δ^i.

Hence, an inference-only path always exists for all non-fundamental triples. $\square$

Now, in our proof of Theorem-1, we have shown the existence of what we call "*trivial sequences*", in which you first compute all of Δ^n before proceeding to Δ^{n+1}. We can further show that non-trivial sequences also exist in which some non-fundamental triples can be computed before all fundamental triples are computed. The proof for this is simpler and hence skipped, but we state this as another theorem:

Theorem 2. *Non-trivial maximal sequences exist.*

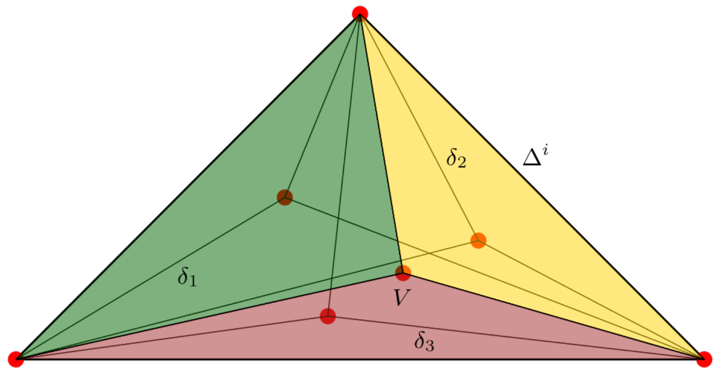


Fig. 3. Theorem-1 Proof Inductive Step: Inferring a Δ^i

Counting Sequences Given that we have proven the existence of trivial and non-trivial maximal sequences, we would now like to attempt a counting for the maximal sequences of both types. This will provide us with a realistic "efficacy" measure of parsimonious algorithms, since we can then compute the probability of hitting the maximal sequences if approached in a random order, and also compute the probable number of inferences we can make (which possibly will be less than the theoretical maximum number of inferences). This counting of sequence types is a challenging problem and beyond the scope of our current work, but we will mention a framework we believe may be useful to approach this problem.

To count inference sequences, we can build a dependency *hypergraph* among triples. A hypergraph, where an edge connects more than two nodes, is a natural fit since our inference process uses three known triples to infer a fourth.

The hypergraph, then, will model each triple of points as a single node. An edge of the hypergraph will connect three nodes to a fourth node, indicating that the fourth triple could be inferred from the first three. A single additional node will be added that represents calculation of the orientation directly, rather than by inference. An edge will be added from this node to all other nodes.

After modeling the problem as such a hypergraph, the problem of finding the orientation for all triples becomes one of finding a spanning tree in this directed hypergraph. The total number of such spanning trees is the total number of ways that we could compute orientations (through combinations of calculation and inference). Further, the spanning trees that minimize the number of edges from the calculation node will be the inference-maximizing options.

4 Experimental and Theoretical Results for Maximal Configuration Analysis

Having discussed the two constructive approaches and existence theorems on inference sequences, we now discuss some of the results we get when we implement these approaches on some point sets.

For our experimental setup, we developed the following programs:

1. A Parsimonious Algorithm implementation to compute all CC relations. The first 3 axioms were included in the data structure of the implementation itself, and axiom-4 was used for parsimonious inferences.
2. A multi-core Brute-Force Search for Maximum Inferences for the above implementation
3. A region enumeration and visualization utility for backward construction

The source code is available at: github.com/shivams/parsimonious-efficacy. All the simulations were run on an 8-core i7 8th generation processor with 32GB RAM.

4.1 Brute-force Search vs Forward Construction

To find the maximizing configuration, we first ran a brute force simulation on a randomly generated set of points, for the 6-point case. The simulation generated random 6-point sets and performed CC computation parsimoniously by choosing a random sequence of triples for its computation. After running the simulation for 3 days on an i7 8th generation 8-core processor (utilizing all cores), we could generate point configurations of up to 5 inferences maximum.

However, when we did forward construction by hand for 6 points, we could obtain as many as 7 inferences (which is the maximum found by our forward construction). This was surprising at first to see that brute-force simulation failed to yield results at merely 6 points, however a quick complexity analysis revealed the explosion of complexity:

- For 6 points, we have a total of 16 point configurations (as shown in Table-1)
- For 6 points, there are $C_3^6 = 20$ triples
- Number of possible sequences of triples: $20! = 2432902008176640000$

The number of possible sequences of triples clearly explodes very quickly. To be specific, the complexity of number of possible sequences comes out to be:

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

And as we have seen in Section-3.2, only certain sequences will yield maximum inferences. Amid this factorial complexity, hitting the inference maximizing sequences becomes much less probable. What decreased probability even further is that there are 16 point configurations, most of which do not admit the maximum number of inferences, and some of which have much lower probability of being realized in a random point set, as has been discussed by [2] (in Chapter-8). This analysis effectively demonstrated that a random search for inference maximizing orientations is not feasible, and that encountering inference maximizing sequences at random is extremely unlikely.

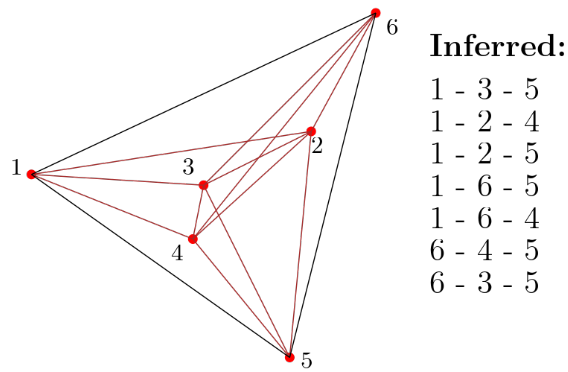


Fig. 4. 6-point Maximal Configuration through (manual) Forward Construction

In Figure-4, we show the maximal configuration for 6-points, which yields 7 inferences. This was arrived at using (manual) forward construction.

4.2 Hypothesizing Inference Maximizing Configuration

We would like to find a configuration of points that is guaranteed to allow maximum inference, and propose two candidate approaches for constructing such configurations. These candidates remain in the hypothesis stage since we do not have a proof yet that these point configurations would yield maximum inferences.

Candidate 1 (Triangle-within-Triangle (TwT)) *In this point configuration, we place points in a triangle. Then, the next set of 3 points are placed in a triangle within it. Then next set within it and so forth, until we exhaust all points.*

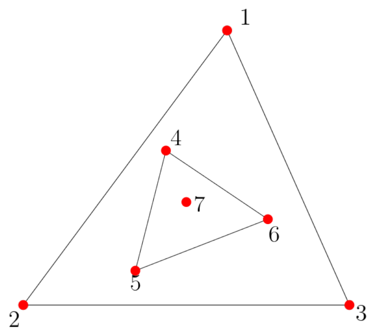
The heuristic that we used to arrive at this configuration goes as follows: We saw in Section 3.1 that the inference minimizing configurations are those in which all points lie on the convex hull. Hence, we do a reverse construction in which we put minimum points on the convex hull (and 3 is the minimum size of a convex hull), and we do that recursively.

Candidate 2 (Spiral) *In this point configuration, we do a forward construction and start placing points in an outward spiral.*

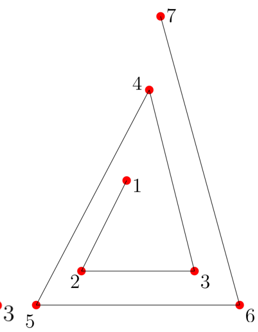
The intuition behind this approach is very similar to that of the triangle-within-triangle approach, except that we are working "outward" rather than

”inward”. Each additional point that is added as we spiral outward ensures that the convex hull remains at only 3 points, and in fact this approach will create triangles within triangles, similar to the prior approach.

Both the configurations are shown in Figure-5. These heuristics of course do not serve as formal proofs and a formal proof remains an open question.



(a) Triangle-within-Triangle



(b) Spiral

Fig. 5. Candidates Hypotheses for Maximal Inferences

5 Conclusion

In our work, we have initiated a study on the “efficacy” of parsimonious algorithms. This is something that has been done for the first time (to the best of our knowledge). We have proved a few simple results, and suggested some initial constructions which we believe will help pave the way for this efficacy analysis. However, implementing these constructions for higher n values becomes challenging. Hence, efficient approaches need to be developed for automating these constructions.

5.1 Open Problems

Our analysis to this point has left us with several open problems, among them:

- **Proving Existence of and Constructing Maximal Configurations**: We have not proved the existence of maximal configurations. We have devised constructive approaches we believe will let us find the maximal configuration, but not having a proof of their existence is a critical gap. Proving their existence or non-existence (or non-decidability about existence) would be a major contribution.
- **Proving the Greedy Property for Constructions**: The constructive approaches take locally optimal choices, but do these choices lead to an overall optimal solution?

– **Tighter Bounds on Empty Triangle Problem**: Constructive approaches for maximal configurations can help shed light on the "empty triangles" problem in Discrete Geometry.
– **Automated Constructions using Order-type Isomorphism**: This is an immediate next step - to completely automate the two constructions we have suggested. The complexity explodes beyond 7 points, but this approach should still be much faster than a brute-force approach. To tackle with complexity explosion, we can enrich this approach with a neural-network based approach [8].
– **Miscellaneous Efficacy Measures**: While we have suggested approaches to analyze maximal configuration and maximal sequences, there are still critical measures we've not quite attained - like the most probable number of inferences one would encounter in a system.

5.2 Extensions

There are also several directions for extending this work that could be useful:

– **Other Axioms**: Our whole analysis has focused on Axiom-4. We could begin to extend the analysis by including the contrapositive of Axiom-4 and/or Axiom-5.
– **Complex Algorithms**: We have focused analysis on only the simplest of geometric problems (computing the order type of planer points). Extending the efficacy analysis of parsimonious algorithms to more sophisticated problems, like Convex Hull and Delaunay Triangulation would be useful.
– **Connecting to Exact and Robust Computation**: Our underlying motivation in this work was to better understand how inference could be used to either support exact computation (by avoiding complex calculation through inference) or improve robustness of inexact geometric computation (by using inference to eliminate possible conflicting predicate evaluations). Making such a connection more explicit would be helpful in understanding how useful inference might or might not be in this process.
– **Optimizing Through Analysis**: This problem of CC Relations has provided a simple case for testing out Parsimonious Algorithms. The concept could possibly be generalized to more general programming problems. That is, we could develop a static analyser which analyses any program, and makes it parsimonious by automatically detecting parts of code which can be inferred from prior knowledge (and hence do not need to be re-computed).
– **Multi-axiom Deduction Analysis**: Finally, we can explore multi-axiom inference, which would be much more useful in real-world situations, but analyzing its efficacy would be even more challenging.
– **Neural-network Assisted Automated Constructions**: This is inspired from the recent work by Google AlphaGeometry[8], in which they had trained a large model to develop an "intuition" for doing constructions for high-school geometry problems. We can extend this approach for doing constructions for our Discrete Geometry problem.

References

1. Aichholzer, O., Aurenhammer, F., Krasser, H.: Enumerating order types for small sets with applications. In: Proceedings of the Seventeenth Annual Symposium on Computational Geometry. pp. 11–18 (2001)
2. Braß, P., Moser, W., Pach, J. (eds.): Research Problems in Discrete Geometry. SpringerLink Bücher, Springer Science+Business Media, Inc, New York, NY (2005). https://doi.org/10.1007/0-387-29929-7
3. Eppstein, D.: Forbidden Configurations in Discrete Geometry:. Cambridge University Press, 1 edn. (May 2018). https://doi.org/10.1017/9781108539180
4. Grünbaum, B.: Configurations of Points and Lines. American Mathematical Soc. (2009)
5. Harborth, H.: Empty triangles in drawings of the complete graph. Discrete Mathematics **191**(1), 109–111 (Sep 1998). https://doi.org/10.1016/S0012-365X(98)00098-3
6. Knuth, D.E.: Axioms and Hulls. No. 606 in Lecture Notes in Computer Science, Springer, Berlin Heidelberg (1992)
7. Pilz, A., Welzl, E.: Order on Order Types. Discrete & Computational Geometry **59**(4), 886–922 (Jun 2018). https://doi.org/10.1007/s00454-017-9912-9
8. Trinh, T.H., Wu, Y., Le, Q.V., He, H., Luong, T.: Solving olympiad geometry without human demonstrations. Nature **625**(7995), 476–482 (Jan 2024). https://doi.org/10.1038/s41586-023-06747-5