Automated Mathematical Discovery and Verification: Minimizing Pentagons in the Plane

Bernardo Subercaseaux ✉, John Mackey, Marijn J.H. Heule, and Ruben Martins

Carnegie Mellon University, Pittsburgh, PA 15213, USA
{bsuberca, jmackey, mheule, rmartins}@andrew.cmu.edu

Abstract. We present a comprehensive demonstration of how automated reasoning can assist mathematical research, both in the discovery of conjectures and in their verification. Our focus is a discrete geometry problem: *What is $\mu_5(n)$, the minimum number of convex pentagons induced by n points in the plane?* In the first stage toward tackling this problem, automated reasoning tools guide discovery and conjectures: we use SAT-based tools to find abstract configurations of points that would induce few pentagons. Afterward, we use Operations Research tools to find and visualize realizations of these configurations in the plane, if they exist. Mathematical thought and intuition are still vital parts of the process for turning the obtained visualizations into general constructions. A surprisingly simple upper bound follows from our constructions: $\mu_5(n) \leq \binom{\lfloor n/2 \rfloor}{5} + \binom{\lceil n/2 \rceil}{5}$, and we conjecture it is optimal. In the second stage, we turn our focus to verifying this conjecture. Using MaxSAT, we confirm that $\mu_5(n)$ matches the conjectured values for $n \leq 16$, thereby improving both the existing lower and upper bounds for $n \in [12, 16]$. Our MaxSAT results rely on two mathematical theorems with pen-and-paper proofs, highlighting once again the rich interplay between automated and traditional mathematics.

Keywords: MaxSAT · Convex Pentagons · Computational Geometry.

1 Introduction

Computation has played an increasingly large role within mathematics over the last 50 years. Back in 1976, Appel and Haken proved the celebrated *Four Color Theorem* using a significant amount of computation [8], which ultimately led to a formally verified Coq proof, written by Georges Gonthier in 2004 [27, 28]. These results serve to highlight a dual role of computing in mathematics: solving problems and verifying solutions [10]. In present times, Large Language Models (LLMs) emerge as a new actor in computer-assisted mathematics; making progress in the *cap-set problem* [40], solving olympiad-level geometry problems [50] and assisting formal theorem proving [53]. Despite claims of progress in AI threatening mathematicians' jobs [19], we adhere to the words of Jordan Ellenberg [16], co-author in the recent *LLMs for the cap-set problem* article [40]:

"What's most exciting to me is modeling new modes of human–machine collaboration, [...] I don't look to use these as a replacement for human mathematicians, but as a force multiplier."

In that spirit, this article presents a self-contained story of human-machine collaboration in mathematics, showcasing how automated reasoning tools can be incorporated in a mathematician's toolkit.

Automated Reasoning. Automated reasoning tools have been successfully used in the past to solve mathematical problems of diverse areas: Erdős Discrepancy Conjecture [35], Keller's conjecture [14], the Packing Chromatic number of the infinite grid [47], and the Pythagorean Triples Problem [31], amongst many others. Interestingly, before the recent progress made with LLMs [40], the prior state of the art for the cap-set problem was obtained via SAT solving [51]. In the context of discrete geometry, Scheucher has used SAT solving to obtain state-of-the-art results in *Erdős-Szekeres* type problems [32, 42, 43], making it our most closely related work. The main novelty of this article is that we apply automated reasoning tools throughout the different stages of a mathematical problem: to guide the discovery of mathematical constructions, elicit a conjecture, and finally verify it until a certain bound to increase our confidence in it.

The Pentagon Minimization Problem. In 1933, Klein presented the following problem [29]: *If five points lie on a plane, without three on a straight line, prove that four of the points will make a convex quadrilateral.* Klein's problem inspired two natural generalizations with a long lasting impact on combinatorial geometry:

Problem 1. For a given $k \geq 3$, is there always a minimum number of points $n = g(k)$, such that any set of n points in the plane, without three in a line, is guaranteed to contain k points that are vertices of a convex k-gon?

Problem 2. For a given $k \geq 3$, what is the minimum number of convex k-gons, $\mu_k(n)$, one can obtain after placing n points in the plane without three in a line?

Erdős and Szekeres published an affirmative answer to Problem 1 in 1935. Szekeres and Klein married shortly afterward, leading Erdős to refer to Problem 1 as the *"Happy Ending Problem"* [38]. Problem 2, on the other hand, is directly mentioned for the first time by Erdős and Guy in 1973 [20]: *"More generally, one can ask for the least number of convex k-gons determined by n points in the plane.* A standard argument we show in Section 6 implies that the limits $c_k := \lim_{n\to\infty} \mu_k(n)/\binom{n}{k}$ are well-defined. Note that $c_3 = 1$ as every set of 3 points in general position forms a triangle. Perhaps surprisingly, c_4 is still unknown despite having received significant attention [1,4]; the best known bounds are roughly $0.3799 \leq c_4 \leq 0.3804$ [4,5]. Computation has played a crucial role in: (i) improving the bounds on c_4, (ii) computing $\mu_k(n)$ for small values of k and n, and (iii) classifying small sets of points according to their geometric relationships [3]. As of today, the value of $\mu_4(n)$ is only known for $n \leq 27$ and $n = 30$ [1],

Table 1: Improvements on $\mu_5(n)$. Brackets indicate the range of values in which $\mu_5(\cdot)$ was known to belong.

# of points (n)	≤ 8	9	10	11	12	13	14	15	16
Previously [2]	0	1	2	7	[12, 13]	[20, 34]	[40, 62]	[60, 113]	—
Our work	0	1	2	7	**12**	**27**	**42**	**77**	**112**

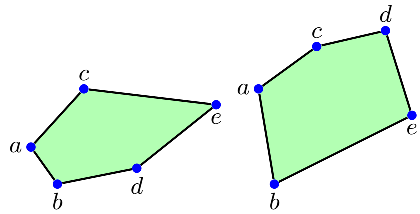
and less is known for $k \geq 5$ [2,3,7,24], where the question raised by Erdős and Guy, 50 years ago, remains widely open. We provide new insights into $\mu_5(n)$ and c_5, and opening directions for studying $\mu_k(n)$ and c_k for larger values of k as well.

Our contributions and outline. This article makes progress on Problem 2 in the particular case of $k = 5$. As shown in Table 1, we fully determine $\mu_5(n)$ for $n \leq 16$, furthering prior results which reached $n = 11$ [2]. We start by providing some background into the geometry of the problem and known values of $\mu_5(n)$ in Section 2, and continue in Section 3 by presenting an initial exploration of $\mu_5(n)$ through *Stochastic Local Search* (SLS) using *signotopes*. Given the absence of open-source programs that find concrete point realizations of signotopes, we present in Section 4 a simple local-search program that finds them. We depict particular realizations obtained for the satisfying assignments that we found through SLS, which provide geometric insight into the problem. Based on those realizations, in Section 5 we propose and study two simple constructions providing a common upper bound of $\mu_5(n) \leq \binom{\lfloor n/2 \rfloor}{5} + \binom{\lceil n/2 \rceil}{5}$, which we conjecture to be optimal. Section 6 shows that if the conjecture holds for an odd value of n, then it will also hold for $n + 1$. Section 7 discusses the verification of the conjecture for $n \leq 16$ through MaxSAT, and finally, Section 8 discusses the impact of the newly found values of $\mu_5(n)$ for bounding c_5 and offers a series of related open problems. Our code is available at https://github.com/bsubercaseaux/minimize-5gons.

2 Background

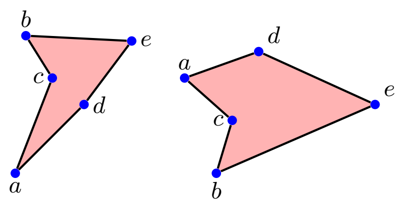
A set of points $S = \{p_1, \ldots, p_n\}$ in the plane (i.e., $p_i = (x_i, y_i) \in \mathbb{R}^2$) is *in general position* when no subset of three points of S belong to a common straight line. Given an integer value n, we aim to find a set S of n points in general position that minimizes the number of convex 5-gons with vertices in S. In order to address this problem computationally one needs to shift from the continuous space $\mathbb{R}^2$ to a discrete and finitely-representable abstraction. The crucial observation to achieve this is that the number of convex pentagons on a set of points does not depend on their exact positions (e.g., it is invariant under scaling or rotations) but rather on the relationship between them.

This is illustrated in Figures 1a and 1b; each of them depicts a pair of pentagons that differ with respect to the exact position of their vertices but are *equiv-*



$\sigma(a,b,c) =$ false	$\sigma(a,d,e) =$ false
$\sigma(a,b,d) =$ false	$\sigma(b,c,d) =$ false
$\sigma(a,b,e) =$ false	$\sigma(b,c,e) =$ false
$\sigma(a,c,d) =$ false	$\sigma(b,d,e) =$ false
$\sigma(a,c,e) =$ false	$\sigma(c,d,e) =$ false

(a) A pair of convex pentagons that are equivalent with respect to their signotopes.



$\sigma(a,b,c) =$ false	$\sigma(a,d,e) =$ false
$\sigma(a,b,d) =$ false	$\sigma(b,c,d) =$ true
$\sigma(a,b,e) =$ true	$\sigma(b,c,e) =$ false
$\sigma(a,c,d) =$ true	$\sigma(b,d,e) =$ false
$\sigma(a,c,e) =$ false	$\sigma(c,d,e) =$ true

(b) A pair of non-convex pentagons that are equivalent with respect to their signotopes.

Fig. 1: Illustration of the signotope abstraction.

alent in terms of the relationships between vertices, in a sense that we make precise next. As it is standard in the area, we abstract a set of points S according to the relative orientation of each of its subsets of three points [34,42,43,48]. We assume without loss of generality that the points are labeled from left to right, that is, for every triple $p_a, p_b, p_c \in S$, with $a < b < c$, we assume $x_a < x_b < x_c$ [48]. Then, for every triple $a < b < c$ we define its *signotope* $\sigma(a, b, c)$ as true if p_a, p_b, p_c appear in counterclockwise order, and false otherwise. Formally,

$$\sigma(a,b,c) = \begin{cases} \mathsf{true} & \text{if } (y_c - y_a)(x_b - x_a) > (x_c - x_a)(y_b - y_a), \\ \mathsf{false} & \text{otherwise.} \end{cases}$$

Importantly, not every combination of signotopes is *consistent* with the left-to-right labeling we assume. For a minimal example, consider points p_a, p_b, p_c, p_d, with $a < b < c < d$, such that p_a, p_b, p_c appear in clockwise order, and p_b, p_c, p_d also appear in clockwise order. Then, necessarily, p_a, p_c, p_d must appear in clockwise order as well, which translates to the implication $\neg\sigma(a,b,c) \wedge \neg\sigma(b,c,d) \implies \neg\sigma(a,c,d)$. More in general, the following *signotope axioms* apply to any set of points in general position, provided that the points are sorted with respect to their x-coordinates [21,48]:

$$\big(\sigma(a,b,c) \vee \neg\sigma(a,b,d) \vee \sigma(a,c,d)\big) \wedge \big(\neg\sigma(a,b,c) \vee \sigma(a,b,d) \vee \neg\sigma(a,c,d)\big) \quad (1)$$
$$\big(\sigma(a,b,c) \vee \neg\sigma(a,c,d) \vee \sigma(b,c,d)\big) \wedge \big(\neg\sigma(a,b,c) \vee \sigma(a,c,d) \vee \neg\sigma(b,c,d)\big) \quad (2)$$
$$\big(\sigma(a,b,c) \vee \neg\sigma(a,b,d) \vee \sigma(b,c,d)\big) \wedge \big(\neg\sigma(a,b,c) \vee \sigma(a,b,d) \vee \neg\sigma(b,c,d)\big) \quad (3)$$
$$\big(\sigma(a,b,d) \vee \neg\sigma(a,c,d) \vee \sigma(b,c,d)\big) \wedge \big(\neg\sigma(a,b,d) \vee \sigma(a,c,d) \vee \neg\sigma(b,c,d)\big) \quad (4)$$

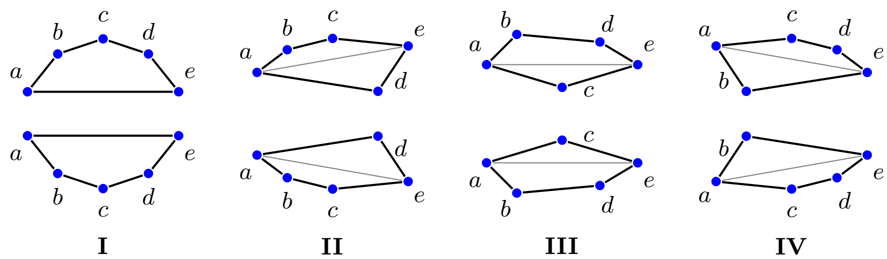


Fig. 2: The convex 5-gon cases based on the position of b, c, d w.r.t. the line $\overline{ae}$.

Even though every set of points in the plane respects these axioms when labeled from left to right, the converse is not true. In other words, there are signotope assignments respecting these axioms that do not correspond to any set of points in the plane; these are said to be *non-realizable*. Deciding whether a set of signotopes is realizable is a well-known hard problem in computational geometry, complete for the class $\exists\mathbb{R}$, and will be discussed further in Section 4.

3 Encoding and Stochastic Local Search

We will use the signotopes $\sigma(a, b, c)$ directly as propositional variables in our encoding. As a first step, we directly add the $O(n^4)$ axiom clauses of Equations (1)-(4). Then, in order to minimize the number of induced convex 5-gons, we use an idea of Szekeres and Peters [48]. Szekeres and Peters identified the four cases that form a convex 5-gon, depending on where the three middle points b, c, and d are located with respect to the line through the leftmost point a and the rightmost point e:

Case I: $\sigma(a, b, c) = \sigma(b, c, d) = \sigma(c, d, e)$
Case II: $\sigma(a, b, c) = \sigma(b, c, e) = \neg\sigma(a, d, e)$
Case III: $\sigma(a, b, d) = \sigma(b, d, e) = \neg\sigma(a, c, e)$
Case IV: $\sigma(a, b, e) = \neg\sigma(a, c, d) = \neg\sigma(c, d, e)$.

The four cases are illustrated in Figure 2, showcasing that each case has two possible orientations depending on the value of the signotopes that the case asserts to be equal. Given that we will first use *Stochastic Local Search* (SLS), it is important to recall that an SLS solver attempts to find an assignment that minimizes the number of falsified clauses in its input formula, without any guarantee of optimality except for when a fully satisfying assignment is found. Therefore, as we want to find assignments to the signotope variables that minimize the number of convex 5-gons, we desire an encoding where each convex 5-gon falsifies exactly one clause. We achieve this by adding the following clauses, based on the disjoint cases **I-IV**, for every sorted tuple of five points (a, b, c, d, e):

$$\sigma(a,b,c) \vee \sigma(b,c,d) \vee \sigma(c,d,e) \quad (5)$$
$$\neg\sigma(a,b,c) \vee \neg\sigma(b,c,d) \vee \neg\sigma(c,d,e) \quad (6)$$
$$\sigma(a,b,c) \vee \sigma(b,c,e) \vee \neg\sigma(a,d,e) \quad (7)$$
$$\neg\sigma(a,b,c) \vee \neg\sigma(b,c,e) \vee \sigma(a,d,e) \quad (8)$$
$$\sigma(a,b,d) \vee \sigma(b,d,e) \vee \neg\sigma(a,c,e) \quad (9)$$
$$\neg\sigma(a,b,d) \vee \neg\sigma(b,d,e) \vee \sigma(a,c,e) \quad (10)$$
$$\sigma(a,b,e) \vee \neg\sigma(a,c,d) \vee \neg\sigma(c,d,e) \quad (11)$$
$$\neg\sigma(a,b,e) \vee \sigma(a,c,d) \vee \sigma(c,d,e). \quad (12)$$

Note that the best assignments found through SLS might, in principle, violate the signotope axiom clauses of Equations (1)-(4) in order to minimize the number of falsified clauses. Interestingly, we tested for about a thousand best assignments whether any axiom clauses were falsified and this was never the case. Therefore the number of falsified clauses in all the best assignments found through SLS, presented in Table 2, would constitute an upper bound on the minimum number of convex 5-gons if they were *realizable*.

We also experimented with formulas without the signotope axiom clauses. The best number of falsified clauses of these formulas match numbers on Table 2. So potentially violating many axiom clauses does not result in fewer 5-gons. It is therefore not surprising that none of the axiom clauses were falsified in the best found assignments. However, the runtimes to obtain the best-known values were substantially higher (roughly an order of magnitude) for the formulas without the axiom clauses. So these clauses are helpful to reduce the runtime.

In terms of software, we tested all the algorithms in UBCSAT [49] and the DDFW algorithm [33] turned out to have the best performance. We ran UBCSAT with its default settings, which means it restarts every 100 000 flips. For some of the harder formulas this resulted in hundreds of restarts. An interesting observation is that the optimal assignments are harder to find when n is even. Observe that best number of falsified clauses matches exactly the conjectured values, apart from $n = 30$ where the best found assignment after 12 hours is 1 above the conjectured value, suggesting that we reached the limit of SLS for this problem.

4 Realizability

Deciding whether a signotope assignment can be realized by a set of points in the plane is a hard combinatorial problem, complete for the complexity class $\exists\mathbb{R}$, which satisfies NP $\subseteq \exists\mathbb{R} \subseteq$ PSPACE [44]. To the best of our knowledge, no open-source tools for the realizability problem are publicly available. Because of this, we present as a contribution of independent interest, a simple local search approach to realizability, that has proved effective in this problem. We use `LocalSolver v12.0` [23], a local search engine that supports floating point

Table 2: SLS results of formulas for $\mu_5(n)$ showing the best number of falsified clauses and the time, in seconds, to find that bound.

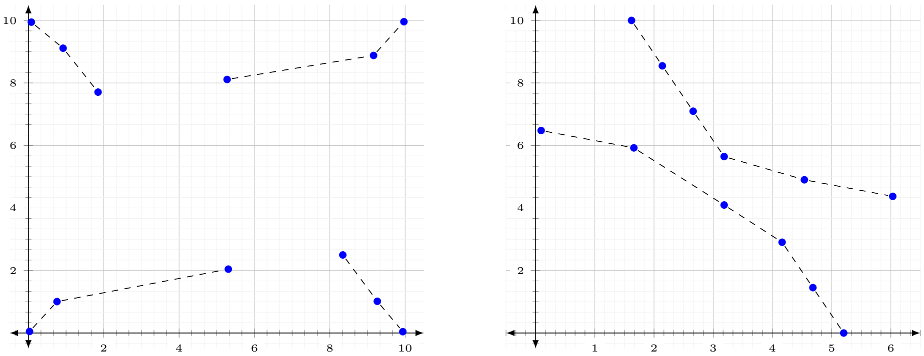
n	9	10	11	12	13	14	15	16	17	18	19	20
best	1	2	7	12	27	42	77	112	182	252	378	504
time [s]	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.02	2.03	0.94	174.11

n	21	22	23	24	25	26	27	28	29	30
best	714	924	1254	1584	2079	2574	3289	4004	5005	6007
time [s]	3.34	43.92	11.64	472.33	63.48	5268.1	1555.5	1791.9	467.36	18 244

variables and provides free academic licenses, in order to find realizations for small numbers of points ($n \leq 16$). Concretely, the problem at hand consists of, given a (not necessarily consistent) assignment to the signotope variables $\sigma(a, b, c), \ldots$, find a set of points $(x_a, y_a), (x_b, y_b), (x_c, y_c), \ldots$ that satisfies them all. We add one constraint per signotope variable, and furthermore maximize the minimum distance between any pair of points, in order to *regularize* the obtained realizations and avoid floating-point-arithmetic issues. Experimentally, we found that if no constraint is placed on the distance between points, `LocalSolver` tends towards solutions that place all points extremely close to each other, or in a straight line, where the constraints will be satisfied only due to floating point imprecision. Therefore, our model consists of:

$$\begin{aligned}
\textbf{maximize} \quad & z \\
\textbf{subject to} \quad & z \leq \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}, \\
& \varepsilon < \sigma^\star(a, b, c) \cdot [(y_c - y_a)(x_b - x_a) - (x_c - x_a)(y_b - y_a)], \\
& 0 \leq x_a \leq K, \\
& 0 \leq y_a \leq K,
\end{aligned}$$

where $\sigma^\star(a, b, c) = 1$ if $\sigma(a, b, c)$ and -1 otherwise. The parameters ε and K are experimentally determined, and also contribute towards avoiding *degenerate* solutions due to floating point arithmetic. In particular, setting a value of ε that is too close to 0 (say, $\varepsilon = 10^{-10}$) allows for degenerate solutions that only satisfy constraints due to floating-point-arithmetic quirks, whereas setting e.g., $\varepsilon = 10^{-3}$ may result in an unfeasible set of constraints even if the signotope assignment is realizable. By implementing this model in `LocalSolver`, we obtained the realizations depicted in Figure 3a and Figure 3b. We remark that, consistently with the asymptotic result stating that most signotope assignments are *not* realizable [13, 44], only about 5% of the 10 000 different assignments we found through SLS for $n \in \{12, 14, 16\}$ led to realizations (in under 100s).



(a) A realization obtained for the problem instance $\mu_5(12) = 12$ that inspires the *pinwheel construction*.

(b) A different realization obtained for the problem instance $\mu_5(12) = 12$ that inspires the *parabolic construction*.

Fig. 3: Illustration of two different realizations obtained with `LocalSolver`. Both realizations come from a single signotope orientation obtained through SLS under different executions. Dashed lines are for illustrative purposes only, in order to showcase the similarity with Figure 4a and Figure 4b.

5 Constructions

We present two different constructions achieving a common bound: (i) the *pinwheel construction* in Section 5.1, which generalizes the realization depicted in Figure 3a, and (ii) the *parabolic construction* in Section 5.2, which generalizes the realization depicted in Figure 3b.

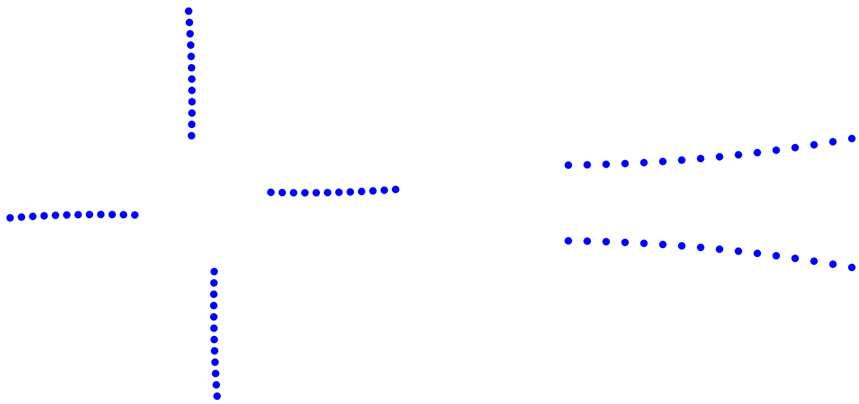
5.1 The Pinwheel Construction

This construction, illustrated in Figure 4a requires the number of points $n = 4k$ to be a multiple of 4. It consists of four *spokes* defined as follows:

$$
\begin{aligned}
S_1 &= \{(k + j, -j(k - j)/k^3 + 1) \mid j \in \{0, 1, \ldots, k - 1\}\} \\
S_2 &= \{(-(k + j), j(k - j)/k^3 - 1) \mid j \in \{0, 1, \ldots, k - 1\}\} \\
S_3 &= \{(j(k - j)/k^3 - 1, k + j) \mid j \in \{0, 1, \ldots, k - 1\}\} \\
S_4 &= \{(-j(k - j)/k^3 + 1, -(k + j)) \mid j \in \{0, 1, \ldots, k - 1\}\}.
\end{aligned}
$$

Proposition 1. *The number of convex* 5*-gons obtained by applying the pinwheel construction on* $n = 4k$ *points is exactly* $2\binom{2k}{5}$.

Proof sketch, illustrated in Figure 5. The proof is by cases according to which spokes contain the 5 points of an arbitrary pentagon. If all five points are contained in the same spoke S_i, then the pentagon is convex due to the curvature of the spokes. If four points are contained in a spoke S_i, then to make a convex



(a) Illustration of the *Pinwheel construction* for $n = 48$ points.

(b) Illustration of the *Parabolic construction* for $n = 32$ points.

Fig. 4: Illustration of the constructions achieving the bound of Theorem 1.

5-gon the fifth point must be in the next spoke counterclockwise, $S_{(i+1)\%4}$. Similarly, if a spoke contains 3 points, and the next spoke counterclockwise contains the remaining 2 points, then the pentagon is convex. Finally, every other case yields a non-convex 5-gon. As there are 4 ways of choosing the unique spoke with the largest number of points in a convex 5-gon, this results in

$$4 \cdot \left(\binom{k}{5} + \binom{k}{4} \cdot \binom{k}{1} + \binom{k}{3} \cdot \binom{k}{2} \right) = 2\binom{2k}{5},$$

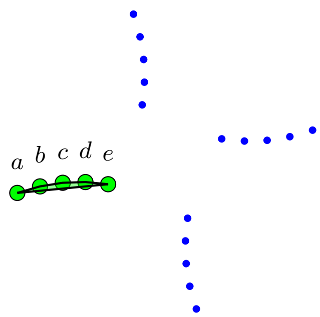
where the equality follows by Vandermonde's identity. □

Proof sketch, illustrated in Figure 6. If we consider any set of five points that are either fully contained in $L^\top := \bigcup_i p_i^\top$ or fully contained in $L^\perp := \bigcup_i p_i^\perp$, then they must define a convex 5-gon, due to the convexity (resp. concavity) of $L^\top$ (resp. $L^\perp$). There are exactly $\binom{\lceil n/2 \rceil}{5} + \binom{\lfloor n/2 \rfloor}{5}$ sets of 5 points that can be chosen in this way. It remains to argue that any 5-gon P that intersects both $L^\top$ and $L^\perp$ cannot be convex. By the pigeonhole principle, there is one curve from $L \in \{L^\perp, L^\top\}$ such that $|L \cap P| \geq 3$, and for this sketch we can assume that $L = L^\perp$ without loss of generality. There are now two cases, illustrated in Figure 6: either $|L^\perp \cap P| = 3$ or $|L^\perp \cap P| = 4$, and both of them lead to non-convex 5-gons due to the concavity of $L^\perp$. □

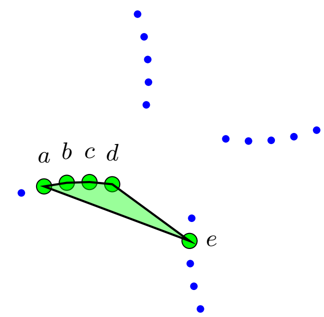
5.2 The Parabolic Construction

A direct generalization of Figure 3b consists on constructing, for any given n:

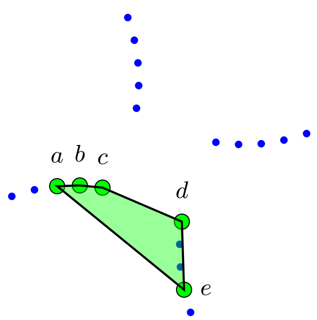
$$p_i^\top = \left(i, 2 + \frac{i^2}{n^2}\right), \forall i \in \left[\left[\left\lfloor \frac{n}{2} \right\rfloor \right]\right] \quad \text{and} \quad p_i^\perp = \left(i, -2 - \frac{i^2}{n^2}\right), \forall i \in \left[\left[\left\lceil \frac{n}{2} \right\rceil \right]\right].$$



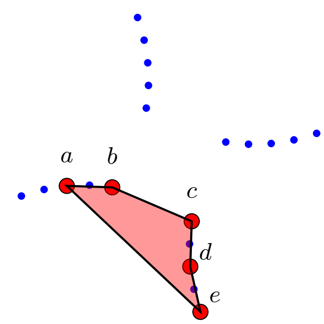
(a) **Case 1:** All five points on the same spoke.



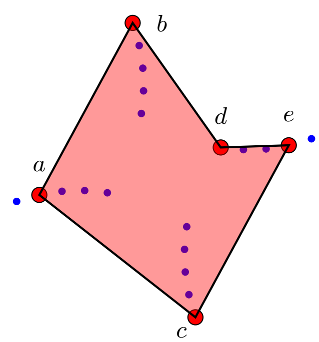
(b) **Case 2:** Four points on a spoke and one in the next spoke counterclockwise.



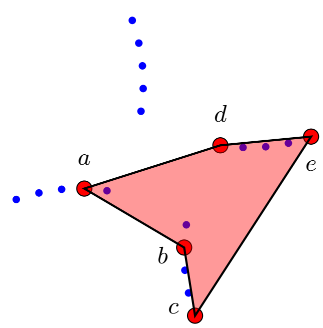
(c) **Case 3:** Three points on a spoke and two on the next spoke counterclockwise.



(d) **Case 4:** Three points on a spoke and two on the next spoke clockwise.

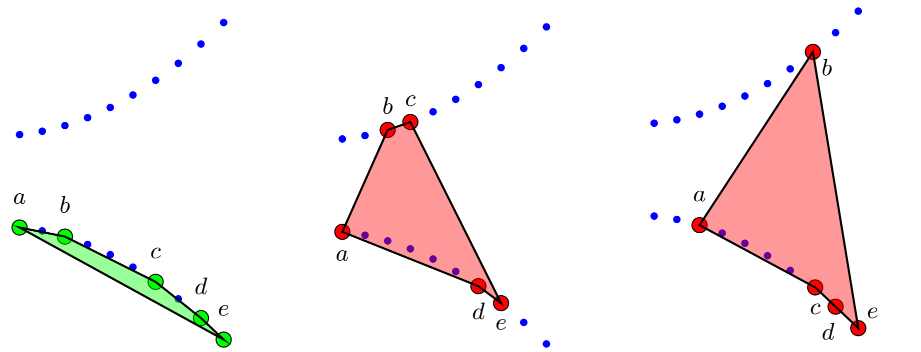


(e) **Case 5:** Four different spokes contain points.



(f) **Case 6:** Three different spokes contain points.

Fig. 5: Illustration for some of the cases in the proof sketch for the pinwheel construction.



(a) **Case 1:** all five points on the same curve necessarily create a convex 5-gon.

(b) **Case 2:** three points on one curve and two on the other never make a convex 5-gon.

(c) **Case 3:** four points on one curve and a single point on the other never make a convex 5-gon.

Fig. 6: Illustration of the proof sketch for the parabolic construction. Curvature of $L^{\top}$ and $L^{\perp}$ has been scaled for illustrative purposes.

An illustration of this construction is presented in Figure 4b. We remark that, albeit having a different use, the parabolic construction seems to be equivalent to the *double chain* idea in [6, 22], but whose application to this problem we discovered thanks to the obtained realizations.

Proposition 2. *The number of convex* 5*-gons obtained by applying the parabolic construction on* n *points is exactly* $\binom{\lfloor n/2 \rfloor}{5} + \binom{\lceil n/2 \rceil}{5}$.

As a direct consequence of the above constructions, we obtain the following upper bound for $\mu_5(n)$.

Theorem 1. *Let* $\mu_5(n)$ *denote the minimum number of convex pentagons when* n *points are placed in the plane in general position. Then,* $\mu_5(n)$ *satisfies the inequality:* $\mu_5(n) \leq \binom{\lfloor n/2 \rfloor}{5} + \binom{\lceil n/2 \rceil}{5}$.

Conjecture 1. The bounds of Theorem 1 are tight.

We remark that albeit the pinwheel construction can be deemed unnecessary as Theorem 1 is already implied by the parabolic construction, we consider it an interesting example of how the diversity of solutions to a MaxSAT problem can translate to a diversity of constructions, or proofs, of a mathematical fact.

6 Odd-Even Implication

Another piece of evidence for Conjecture 1 is given by the following pen-and-paper theorem: if the conjecture holds for $2n-1$ points, then it must hold for $2n$ points.

Proposition 3. *If for some $n > 5$ it holds that $\mu_5(2n-1) = \binom{n}{5} + \binom{n-1}{5}$, then $\mu_5(2n) = 2\binom{n}{5}$.*

In order to prove this, we will generalize the following folklore idea [15, 41]: $\mu_5(n) \geq \frac{n}{n-5}\mu_5(n-1)$, for any n.

Lemma 1. *Let m and r be values such that $\mu_k(m) \geq r$. Then for every $n \geq m$ we have $\mu_k(n) \geq r \cdot \binom{n}{m}/\binom{n-k}{m-k} = r \cdot \binom{n}{k}/\binom{m}{k}$.*

Proof sketch. For each of the $\binom{n}{m}$ subsets of m points, we know there will be at least r convex k-gons. However, this will count multiple times a fixed convex k-gon that appears in many m-point subsets. In particular, each convex k-gon will be counted $\binom{n-k}{m-k}$ times this way, thus yielding the first inequality. The equality comes simply from:

$$\begin{aligned}\binom{n}{m}/\binom{n-k}{m-k} &= \frac{n!}{m!\,(n-m)!} \cdot \frac{(m-k)!\,(n-m)!}{(n-k)!} \\ &= \frac{n!}{(n-k)!\,k!} \cdot \frac{(m-k)!\,k!}{m!} = \binom{n}{k}/\binom{m}{k}. \square\end{aligned}$$

Corollary 1. *The limits $c_k := \lim_{n\to\infty} \mu_k(n)/\binom{n}{k}$ are well defined.*

Proof. Use Lemma 1 with $m = n-1$ and $r = \mu_k(n-1)$, to get

$$\mu_k(n)/\binom{n}{k} \geq \mu_k(n-1)/\binom{n-1}{k},$$

which implies the sequence $\mu_k(n)/\binom{n}{k}$ is non-decreasing, and as it is clearly bounded above by 1, we conclude. □

Moreover, using Lemma 1 with $k = 5$, $m = n-1$, and $r = \mu_k(n-1)$ yields:

Corollary 2. *For any $n > 5$, $\mu_5(n) \geq \frac{n}{n-5} \cdot \mu_5(n-1)$.*

We are now ready to prove Proposition 3.

Proof of Proposition 3. By using Theorem 1 we have $\mu_5(2n) \leq 2\binom{n}{5}$. Now, to argue that equality is achieved, we use Corollary 2 to obtain that

$$\begin{aligned}\mu_5(2n) \geq \frac{2n}{2n-5} \cdot \mu_5(2n-1) &= \frac{2n}{2n-5}\left(\binom{n}{5} + \binom{n-1}{5}\right) \\ &= \binom{n}{5} + \frac{5}{2n-5}\binom{n}{5} + \frac{2 \cdot n! \cdot (n-5)}{(2n-5)(n-5)!\,5!} \\ &= \binom{n}{5} + \frac{5}{2n-5}\binom{n}{5} + \frac{2(n-5)}{2n-5}\binom{n}{5} = 2\binom{n}{5}.\end{aligned}$$

The two inequalities imply that $\mu_5(2n) = 2\binom{n}{5}$ and conclude the proof. □

Table 3: Number of variables (#Vars), hard clauses (#Hard), soft clauses (#Soft), and symmetry breaking clauses (#Symmetry).

Instance	#Vars	#Hard	#Softs	#Symmetry
$\mu_5(9)$	210	2 016	126	28
$\mu_5(11)$	627	6 336	462	45
$\mu_5(13)$	1 573	16 016	1 287	66
$\mu_5(15)$	3 458	34 944	3 003	91

7 MaxSAT Verification

The Stochastic Local Search presented in Section 3 gives us upper bounds on the value of $\mu_5(n)$ for specific values of n, on the condition that the associated signotope assignments are realizable. Thanks to Theorem 1, obtained through the constructions of Section 5, we know the upper bounds found through SLS are indeed true upper bounds for $\mu_5(n)$. However, it could be the case that better bounds could be found since Stochastic Local Search does not provide any guarantees of optimality. To further support our conjecture, we will use Maximum Satisfiability (MaxSAT) [11] solvers to find the *optimal value* of $\mu_5(n)$ for some values of n. In particular, we show that $\mu_5(9) = 1$, $\mu_5(11) = 7$, $\mu_5(13) = 27$, and $\mu_5(15) = 77$. Concretely, as every set of points in the plane is captured by the signotope abstraction, if no assignment of signotopes induces fewer than m convex 5-gons for a given value of n, then indeed we conclude $\mu_5(n) \geq m$.

7.1 MaxSAT Encoding

MaxSAT is an optimization variant of SAT, where, given an unsatisfiable formula, the goal is to maximize the number of satisfied clauses. MaxSAT can be extended to include two sets of clauses: *hard* and *soft*. An optimal assignment for a MaxSAT problem satisfies all hard clauses while maximizing the number of satisfied soft clauses. To build a MaxSAT encoding for this problem we first introduce *relaxation* variables $r(a, b, c, d, e)$ to denote whether the 5-gon with vertices a, b, c, d, and e is convex and thus must be avoided. We then modify clauses (5)-(12) by adding the literal $r(a, b, c, d, e)$ to each of them.

The axiom clauses (1)-(4) in Section 2 and the modified clauses (5)-(12) are defined as being hard. Finally we introduce the following soft clauses:

$$\bigwedge_{(a,b,c,d,e)\in S} \big(\neg r(a, b, c, d, e)\big). \tag{13}$$

The MaxSAT formulas for $\mu_5(n)$ are modest in size for small n, with $\mu_5(15)$ featuring about 3 500 variables, 35 000 hard clauses, and 3 000 soft clauses. Table 3 shows the size of the MaxSAT formula for $\mu_5(n)$ (including symmetry-breaking clauses described next in Section 7.1).

Table 4: Experimental results **without symmetry-breaking constraints**: Wall clock time in seconds to solve $\mu_5(n)$ with a time limit of 18,000 seconds (5 hours) per instance. A '–' denotes a timeout was reached, and optimality was not proven. For the cube-and-conquer approach (C&C), we also include in parenthesis the sum of the CPU time needed to solve all disjoint formulas.

	Solver	$\mu_5(9)$	$\mu_5(11)$	$\mu_5(13)$	$\mu_5(15)$
Sequential	EvalMaxSAT	5.01	485.08	–	–
	MaxCDCL	**0.02**	35.28	–	–
	Pacose	0.02	99.69	–	–
	MaxHS	0.03	–	–	–
C&C	EvalMaxSAT	4.98 (39.40)	287.27 (3,728.45)	–	–
	MaxCDCL	0.02 (0.14)	**2.27** (19.42)	**1,004.19** (18,133.54)	–
	Pacose	0.02 (0.14)	5.06 (61.11)	–	–
	MaxHS	0.02 (0.17)	–	–	–

Symmetry Breaking. Adding symmetry-breaking predicates that remove equivalent solutions can prune the search space and improve the performance of SAT solvers [18, 37] and is beneficial as well for MaxSAT solvers. For $\mu_5(n)$, we can break some symmetries by adding hard unit clauses $\sigma(1, b, c)$ with $b < c$ so that only solutions where points appear in counterclockwise order with respect to p_1 are obtained. A proof of correctness for this symmetry breaking is presented in [42, Lemma 1]. By comparing Tables 3 to 5, we observe that adding just a few unit clauses to break symmetries has a significant impact on MaxSAT solver performance. For instance, $\mu_5(15)$ is unsolvable with any approach, compared to a 32-minute solve time with MaxCDCL (see Table 5 in Section 7). For $\mu_5(13)$, MaxCDCL with cube-and-conquer and symmetry-breaking predicates solves it in 7.69 seconds, compared to over 1 000 seconds without symmetry breaking. This effect is also seen with other solvers, highlighting the importance of symmetry breaking in practical problem-solving.

MaxSAT Approaches. MaxSAT solvers employ various strategies for finding optimal solutions. This paper explores four MaxSAT solvers (EvalMaxSAT, MaxCDCL, Pacose, and MaxHS) that have excelled in the annual MaxSAT Evaluations.[1] EvalMaxSAT [9] uses an unsatisfiability-based algorithm, beginning with a linear search from the lower bound to the optimal solution. The winning version in the MaxSAT Evaluation 2023 used an integer linear programming (ILP) solver as a preprocessing step but performed better without it in our evaluation. MaxCDCL [36] combines clause learning with branch-and-bound and was among the top-performing solvers in the MaxSAT Evaluation 2023. We used the default version without ILP preprocessing. Pacose [39] performs a linear search, iteratively improving the upper bound until it finds an optimal solution. In our evaluation, we utilized the version from the MaxSAT Evaluation 2021. MaxHS [17] employs an implicit hitting set approach, combining SAT and ILP

[1] https://maxsat-evaluations.github.io/

solvers. It was the leading solver in the MaxSAT Evaluation 2021 and continues to excel in solving MaxSAT problems.

Cube-and-conquer. We explore a strategy inspired by the cube-and-conquer approach to parallelizing SAT formulas [30]. The key idea behind cube-and-conquer is to split a formula into 2^n disjoint formulas by carefully choosing n variables and fixing their truth values to all possible 2^n combinations. For instance, consider Boolean variables x_1 and x_2 that belong to a formula φ. This formula can be split into 4 disjoint formulas with the following construction $\varphi_1 = \varphi \cup (x_1 \wedge x_2)$, $\varphi_2 = \varphi \cup (\neg x_1 \wedge x_2)$, $\varphi_3 = \varphi \cup (x_1 \wedge \neg x_2)$, and $\varphi_4 = \varphi \cup (\neg x_1 \wedge \neg x_2)$. The intuition behind this idea is that it is easier to solve φ_i than φ and that this approach can be used to create many disjoint formulas and enable massive parallelism. In our evaluation, we selected the variables $\sigma(3,4,5)$, $\sigma(5,6,7)$, $\ldots$, $\sigma(n-2, n-1, n)$ (3 for $\mu_5(9)$, 4 for $\mu_5(11)$, 5 for $\mu_5(13)$, and 6 for $\mu_5(15)$), because we experimentally observed that these variables split the search into balanced subspaces. Note that an optimal solution for φ corresponds to the best optimal solution for *all* φ_i. Even though this is just a preliminary study on using the cube-and-conquer approach to solve MaxSAT formulas, we show that even a few variables can have a significant impact on allowing us to solve harder problems.

Finding Optimal Values for $\mu_5(n)$. We run EvalMaxSAT, MaxCDCL, Pacose, and MaxHS with the sequential and cube-and-conquer versions on the StarExec cluster [45] −Intel(R) Xeon(R) CPU E5-2609 @ 2.40GHz−with a memory limit of 32 GB. All experiments were run with a time limit of 5 hours (wall-clock time) per benchmark (which is the largest time limit allowed by StarExec). Symmetry-breaking predicates were applied to all formulas, as they are crucial for effective problem-solving (cf. Tables 4 and 5). Table 5 shows that MaxSAT can be used to prove the optimality of small values of n for $\mu_5(n)$. Note that the best exact values for $\mu_5(n)$ prior to this work were up to $\mu_5(11)$. By using MaxSAT we can improve the best-known bounds for $\mu_5(n)$ up to $n = 16$.[2] Determining $\mu_5(n)$ is a challenging problem for current MaxSAT tools, and while all of the evaluated tools could solve $\mu_5(11)$, only MaxCDCL was able to solve $\mu_5(13)$ using the sequential version and $\mu_5(15)$ with the cube-and-conquer approach. To the best of our knowledge, this is the first example of how cube-and-conquer can improve the performance of MaxSAT solvers. For instance, MaxHS can solve $\mu_5(11)$ in 28.45, seconds while it would take 15 times more wall clock time to solve it using the sequential approach. Even when considering the sum of CPU taken by all disjoint formulas, it is still beneficial to use cube-and-conquer for most cases. The cube-and-conquer approach can leverage having multiple machines available on a cluster, such as StarExec, to improve the scalability of MaxSAT tools, and allows EvalMaxSAT and Pacose to solve $\mu_5(13)$ within the allocated time budget. Furthermore, it improved the performance of MaxCDCL to solve $\mu_5(15)$ in approximately 32 minutes. These results encourage further exploration

[2] While we have exact values only up to $n = 15$, the odd-even implication (see Section 6) guarantees that the conjecture must also hold for $n = 16$.

of a cube-and-conquer approach to solve other hard combinatorial problems with MaxSAT, and open new research directions on how to automatically select splitting variables for creating disjoint subformulas in the context of MaxSAT.

Table 5: Experimental results **with symmetry-breaking constraints**: Wall clock time in seconds to solve $\mu_5(n)$ with a time limit of 18 000 seconds (5 hours) per instance. A '−' denotes a timeout was reached, and optimality was not proven. For the cube-and-conquer approach (C&C), we also include in parenthesis the sum of the CPU time needed to solve all disjoint formulas.

	Solver	$\mu_5(9)$	$\mu_5(11)$	$\mu_5(13)$	$\mu_5(15)$
Sequential	EvalMaxSAT	3.39	237.59	−	−
	MaxCDCL	**0.02**	0.49	150.59	−
	Pacose	0.03	1.93	−	−
	MaxHS	0.03	426.97	−	−
C&C	EvalMaxSAT	0.92 (5.05)	96.65 (873.05)	825.11 (22 015.37)	−
	MaxCDCL	0.03 (0.46)	**0.15** (1.46)	**7.69** (140.88)	**1 930.40** (66 333.04)
	Pacose	0.03 (0.46)	0.19 (1.73)	136.70 (2 647.45)	−
	MaxHS	0.03 (0.44)	28.45 (93.32)	−	−

Certification of Results. Unlike SAT competitions, where SAT solvers provide proofs of unsatisfiability that can be independently verified, MaxSAT solvers in the MaxSAT Evaluation do not offer proofs of optimality. Consequently, there is a possibility of incorrect results. Recently, certain MaxSAT techniques [12,25,52] have emerged, capable of generating verifiable proofs of optimality using the verifier VeriPB [26]. We used the VeritasPBLib [25] framework to generate a certified CNF formula that encodes the $\mu_5(n)$ SAT problem with an additional constraint that enforces the bound to be smaller than our conjectured best value. This is similar to what Pacose does in the last step of its search algorithm when it proves that the last solution found is optimal. We can feed the resulting formula to a SAT solver and verify the unsatisfiability proof with VeriPB. This approach can solve $\mu_5(9)$ and $\mu_5(11)$ and verify both results within a few seconds. Unfortunately, larger values of n are beyond the reach of this approach since this construction (like Pacose) cannot solve $\mu_5(13)$ within the 5-hour time limit.

8 Concluding Remarks and Future Work

We have proved the upper bound of Theorem 1 in two different ways by exhibiting two different constructions, and verified through MaxSAT that this bound is tight at least up to $n = 16$. Moreover, we have proven that the conjecture cannot fail for the first time at an even number of points. Following the tradition of Erdős, we offer a reward of $500 for the first person to prove or disprove Conjecture 1.

On the Constant c_5. Let us now discuss bounds on the constant c_5 that our work implies. First, we note that Theorem 1 provides an upper bound to c_5 as follows. We will use the equation $\lim_{n\to\infty} \binom{n}{k}/n^k = \frac{1}{k!}$, which holds for any fixed integer $k > 0$. Now, if we consider the subsequence of even numbers $2n$, we have

$$c_5 = \lim_{2n\to\infty} \frac{\mu_5(2n)}{\binom{2n}{5}} \leq \lim_{2n\to\infty} \frac{2\binom{n}{5}}{\binom{2n}{5}} = \lim_{2n\to\infty} \frac{2 \cdot n^5 \cdot 5!}{(2n)^5 \cdot 5!} = \frac{1}{16} = 0.0625.$$

After we had written this article, it has come to our attention that this upper bound appears in the work of Goaoc et al. [24], however, they do not provide proof. Nonetheless, their work provides a strong lower bound of $c_5 \geq 0.0608516$. Improving on this lower bound through SAT solving seems very challenging, as we show next. We know that $\mu_5(16) = 112$, from where Lemma 1 yields that for $n > 16$ we have

$$\mu_5(n) \geq 112 \cdot \frac{\binom{n}{16}}{\binom{n-5}{11}} = 112 \cdot \binom{n}{5} / \binom{16}{5} = \frac{112}{4368} \cdot \binom{n}{5},$$

from where $c_5 \geq \frac{112}{4368} \approx 0.02564$. Following the same method for $n = 380$ yields $c_5 \geq 0.060857$. That is, improving on the bound of Goaoc et al. [24] would require solving $n = 380$, which is currently out of reach.

Open Problems. We offer the following challenges:

1. **($500)** Prove or disprove Conjecture 1.
2. Obtain and verify the value of $\mu_5(17)$ and $\mu_5(19)$. If those values also match our conjectured bounds (182 and 378), this would contribute with an even stronger piece of evidence for our conjecture; the first 6 odd values of n would fit the degree 5 polynomial we proposed for odd n, meaning that if $\mu_5(2n + 1)$ were to be a fixed polynomial of degree 5, then it must be the one we proposed.
3. Obtain concrete bounds and design constructions for $\mu_6(\cdot)$ and $\mu_7(\cdot)$. In particular, we have checked that the parabolic construction is not optimal for $\mu_6(\cdot)$, thus suggesting that new insights will be needed. We hope our methodology based on realizations of SLS results can be helpful in this case as well.
4. It is well known by now that symmetry breaking can provide dramatic performance advantages for SAT-solving in combinatorial problems [18,37,46,47]. A natural question is whether the 4-fold symmetry of the pinwheel construction can be assumed without loss of generality, and if so, what would be an efficient way of taking advantage of that fact. It is worth mentioning here that the threefold symmetry of constructions for $\mu_4(\cdot)$ has been repeatedly conjectured to achieve optimality [1], and yet remains unproven.

Acknowledgments. This work is partially supported by the U.S. National Science Foundation (NSF) under grant CCF-2108521. We thank anonymous reviewers from CICM 2024 for their feedback and suggestions.

References

1. Ábrego, B.M., Fernández-Merchant, S., Salazar, G.: The rectilinear crossing number of k_n : Closing in (or are we?). In: Pach, J. (ed.) Thirty Essays on Geometric Graph Theory. pp. 5–18. Springer New York, New York, NY (2013). https://doi.org/10.1007/978-1-4614-0110-0_2
2. Aichholzer, O.: [empty][colored] k-gons - Recent results on some Erdős-Szekeres type problems. Proc. EGC2009 pp. 43–52 (2009)
3. Aichholzer, O., Aurenhammer, F., Krasser, H.: Enumerating order types for small sets with applications. In: Proceedings of the seventeenth annual symposium on Computational geometry. p. 11–18. SCG '01, Association for Computing Machinery, New York, NY, USA (Jun 2001). https://doi.org/10.1145/378583.378596
4. Aichholzer, O., Duque, F., Fabila-Monroy, R., García-Quintero, O.E., Hidalgo-Toscano, C.: An ongoing project to improve the rectilinear and the pseudolinear crossing constants. Journal of Graph Algorithms and Applications **24**(3), 421–432 (2020). https://doi.org/10.7155/jgaa.00540
5. Aichholzer, O., Duque, F., Fabila-Monroy, R., Hidalgo-Toscano, C., García-Quintero, O.E.: An ongoing project to improve the rectilinear and the pseudolinear crossing constants (Jul 2019), https://arxiv.org/abs/1907.07796v5
6. Aichholzer, O., Fabila-Monroy, R., González-Aguilar, H., Hackl, T., Heredia, M.A., Huemer, C., Urrutia, J., Valtr, P., Vogtenhuber, B.: On k-gons and k-holes in point sets. Computational Geometry **48**(7), 528–537 (Aug 2015). https://doi.org/10.1016/j.comgeo.2014.12.007
7. Aichholzer, O., Hackl, T., Vogtenhuber, B.: On 5-Gons and 5-Holes, p. 1–13. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34191-5_1
8. Appel, K., Haken, W.: The Four-Color Problem. In: Steen, L.A. (ed.) Mathematics Today Twelve Informal Essays, pp. 153–180. Springer New York, New York, NY (1978). https://doi.org/10.1007/978-1-4613-9435-8_7
9. Avellaneda, F.: Evalmaxsat 2023. MaxSAT Evaluation 2023 p. 12 (2023)
10. Avigad, J.: Mathematics and the formal turn. Bulletin of the American Mathematical Society **61**(2), 225–240 (Feb 2024). https://doi.org/10.1090/bull/1832
11. Bacchus, F., Järvisalo, M., Martins, R.: Maximum satisfiabiliy. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability - Second Edition, Frontiers in Artificial Intelligence and Applications, vol. 336, pp. 929–991. IOS Press (2021)
12. Berg, J., Bogaerts, B., Nordström, J., Oertel, A., Vandesande, D.: Certified core-guided maxsat solving. In: International Conference on Automated Deduction. pp. 1–22. Springer (2023)
13. Björner, A., Las Vergnas, M., Sturmfels, B., White, N., Ziegler, G.M.: Oriented Matroids. Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, 2 edn. (1999). https://doi.org/10.1017/CBO9780511586507
14. Brakensiek, J., Heule, M., Mackey, J., Narváez, D.: The resolution of keller's conjecture (2023)
15. Brodsky, A., Durocher, S., Gethner, E.: The rectilinear crossing number of k_{10} is 62. The Electronic Journal of Combinatorics p. R23–R23 (Apr 2001). https://doi.org/10.37236/1567
16. Castelvecchi, D.: DeepMind AI outdoes human mathematicians on unsolved problem. Nature **625**(7993), 12–13 (Dec 2023). https://doi.org/10.1038/d41586-023-04043-w

17. Davies, J., Bacchus, F.: Postponing optimization to speed up MAXSAT solving. In: Schulte, C. (ed.) Principles and Practice of Constraint Programming - 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8124, pp. 247–262. Springer (2013). https://doi.org/10.1007/978-3-642-40627-0_21
18. Devriendt, J., Bogaerts, B., Bruynooghe, M., Denecker, M.: Improved static symmetry breaking for SAT. In: Creignou, N., Berre, D.L. (eds.) Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference. Lecture Notes in Computer Science, vol. 9710, pp. 104–122. Springer (2016)
19. Eloundou, T., Manning, S., Mishkin, P., Rock, D.: Gpts are gpts: An early look at the labor market impact potential of large language models (2023)
20. Erdős, P., Guy, R.K.: Crossing number problems. The American Mathematical Monthly **80**(1), 52–58 (1973). https://doi.org/10.2307/2319261
21. Felsner, S., Weil, H.: Sweeps, arrangements and signotopes. Discrete Applied Mathematics **109**(1), 67–94 (Apr 2001). https://doi.org/10.1016/S0166-218X(00)00232-8
22. García, A., Noy, M., Tejel, J.: Lower bounds on the number of crossing-free subgraphs of K_n. Computational Geometry **16**(4), 211–221 (Aug 2000). https://doi.org/10.1016/S0925-7721(00)00010-9
23. Gardi, F., Benoist, T., Darlay, J., Estellon, B., Megel, R.: Mathematical Programming Solver Based on Local Search. Wiley (Jun 2014). https://doi.org/10.1002/9781118966464
24. Goaoc, X., Hubard, A., de Joannis de Verclos, R., Sereni, J.S., Volec, J.: Limits of Order Types. In: Arge, L., Pach, J. (eds.) 31st International Symposium on Computational Geometry (SoCG 2015). Leibniz International Proceedings in Informatics (LIPIcs), vol. 34, pp. 300–314. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2015). https://doi.org/10.4230/LIPIcs.SOCG.2015.300
25. Gocht, S., Martins, R., Nordström, J., Oertel, A.: Certified CNF translations for pseudo-boolean solving. In: Meel, K.S., Strichman, O. (eds.) 25th International Conference on Theory and Applications of Satisfiability Testing, SAT 2022. LIPIcs, vol. 236, pp. 16:1–16:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.SAT.2022.16
26. Gocht, S., Nordström, J.: Certifying parity reasoning efficiently using pseudo-Boolean proofs. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI '21). pp. 3768–3777 (Feb 2021)
27. Gonthier, G.: The Four Colour Theorem: Engineering of a Formal Proof. In: Kapur, D. (ed.) Computer Mathematics. pp. 333–333. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
28. Gonthier, G.: A computer-checked proof of the Four Color Theorem. Tech. rep., Inria (Mar 2023), https://inria.hal.science/hal-04034866
29. Graham, R.L., Spencer, J.H.: Ramsey theory. Scientific American **263**(1), 112–117 (1990)
30. Heule, M.J.H., Kullmann, O., Biere, A.: Cube-and-conquer for satisfiability. In: Hamadi, Y., Sais, L. (eds.) Handbook of Parallel Constraint Reasoning, pp. 31–59. Springer (2018). https://doi.org/10.1007/978-3-319-63516-3_2
31. Heule, M.J.H., Kullmann, O., Marek, V.W.: Solving and Verifying the Boolean Pythagorean Triples Problem via Cube-and-Conquer, p. 228–245. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-40970-2_15
32. Heule, M.J.H., Scheucher, M.: Happy Ending: An Empty Hexagon in Every Set of 30 Points. In: Tools and Algorithms for the Construction and Analysis of Systems

- 30th International Conference, TACAS 2024, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings, Part I. pp. 61–80 (2024). https://doi.org/10.1007/978-3-031-57246-3_5
33. Ishtaiwi, A., Thornton, J., Sattar, A., Pham, D.N.: Neighbourhood clause weight redistribution in local search for sat. In: van Beek, P. (ed.) Principles and Practice of Constraint Programming - CP 2005. pp. 772–776. Springer (2005)
34. Knuth, D.E.: Axioms and Hulls, p. 1–98. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (1992). https://doi.org/10.1007/3-540-55611-7_1
35. Konev, B., Lisitsa, A.: A sat attack on the erdos discrepancy conjecture (2014)
36. Li, C., Xu, Z., Coll, J., Manyà, F., Habet, D., He, K.: Combining clause learning and branch and bound for maxsat. In: Michel, L.D. (ed.) 27th International Conference on Principles and Practice of Constraint Programming. LIPIcs, vol. 210, pp. 38:1–38:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)
37. Metin, H., Baarir, S., Colange, M., Kordon, F.: Cdclsym: Introducing effective symmetry breaking in SAT solving. In: Beyer, D., Huisman, M. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference, TACAS 2018, Held as Part of ETAPS, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10805, pp. 99–114. Springer (2018)
38. Morris, W., Soltan, V.: The Erdős-Szekeres problem on points in convex position – a survey. Bulletin of the American Mathematical Society **37**(4), 437–458 (Jun 2000). https://doi.org/10.1090/S0273-0979-00-00877-6
39. Paxian, T., Reimer, S., Becker, B.: Pacose: An iterative sat-based maxsat solver. MaxSAT Evaluation **2018**, 20 (2018)
40. Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M.P., Dupont, E., Ruiz, F.J.R., Ellenberg, J.S., Wang, P., Fawzi, O., Kohli, P., Fawzi, A.: Mathematical discoveries from program search with large language models. Nature **625**(7995), 468–475 (Jan 2024). https://doi.org/10.1038/s41586-023-06924-6
41. Scheinerman, E.R., Wilf, H.S.: The rectilinear crossing number of a complete graph and Sylvester's "four point problem" of geometric probability. The American Mathematical Monthly **101**(10), 939–943 (1994). https://doi.org/10.2307/2975158
42. Scheucher, M.: Two disjoint 5-holes in point sets. Computational Geometry **91** (Dec 2020). https://doi.org/10.1016/j.comgeo.2020.101670
43. Scheucher, M.: A sat attack on Erdős-Szekeres numbers in r^d and the empty hexagon theorem. Computing in Geometry and Topology **2**(1), 2:1–2:13 (Mar 2023). https://doi.org/10.57717/cgt.v2i1.12
44. Shor, P.W.: Stretchability of pseudolines is NP-hard. In: Gritzmann, P., Sturmfels, B. (eds.) Applied Geometry And Discrete Mathematics, Proceedings of a DIMACS Workshop, Providence, Rhode Island, USA, September 18, 1990. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 4, pp. 531–554. DIMACS/AMS (1990). https://doi.org/10.1090/dimacs/004/41
45. Stump, A., Sutcliffe, G., Tinelli, C.: Starexec: A cross-community infrastructure for logic solving. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) Automated Reasoning - 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014. Lecture Notes in Computer Science, vol. 8562, pp. 367–373. Springer (2014). https://doi.org/10.1007/978-3-319-08587-6_28
46. Subercaseaux, B., Heule, M.: Toward optimal radio colorings of hypercubes via sat-solving. In: Piskac, R., Voronkov, A. (eds.) Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning. EPiC Series in Computing, vol. 94, pp. 386–404. EasyChair (2023). https://doi.org/10.29007/qrmp

47. Subercaseaux, B., Heule, M.J.H.: The packing chromatic number of the infinite square grid is 15. In: Sankaranarayanan, S., Sharygina, N. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 29th International Conference, TACAS 2023, Held as Part of ETAPS 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13993, p. 389–406. Springer (2023). https://doi.org/10.1007/978-3-031-30823-9_20
48. Szekeres, G., Peters, L.: Computer solution to the 17-point Erdős-Szekeres problem. The ANZIAM Journal **48**(2), 151–164 (2006). https://doi.org/10.1017/S144618110000300X
49. Tompkins, D.A.D., Hoos, H.H.: UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT. In: Hoos, H.H., Mitchell, D.G. (eds.) Theory and Applications of Satisfiability Testing. pp. 306–320. Springer, Berlin, Heidelberg (2005)
50. 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
51. Tyrrell, F.: New Lower Bounds for Cap Sets (Dec 2023). https://doi.org/10.19086/da.91076
52. Vandesande, D., De Wulf, W., Bogaerts, B.: Qmaxsatpb: A certified maxsat solver. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 429–442. Springer (2022)
53. Yang, K., Swope, A., Gu, A., Chalamala, R., Song, P., Yu, S., Godil, S., Prenger, R., Anandkumar, A.: LeanDojo: Theorem proving with retrieval-augmented language models. In: Neural Information Processing Systems (NeurIPS) (2023)