Convex Polygon Containment: Improving Quadratic to Near Linear Time

Timothy M. Chan
Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA

Isaac M. Hair
Department of Computer Science, University of California, Santa Barbara, CA, USA

Abstract

We revisit a standard polygon containment problem: given a convex k-gon P and a convex n-gon Q in the plane, find a placement of P inside Q under translation and rotation (if it exists), or more generally, find the largest copy of P inside Q under translation, rotation, and scaling.

Previous algorithms by Chazelle (1983), Sharir and Toledo (1994), and Agarwal, Amenta, and Sharir (1998) all required $\Omega(n^2)$ time, even in the simplest $k = 3$ case. We present a significantly faster new algorithm for $k = 3$ achieving $O(n \operatorname{polylog} n)$ running time. Moreover, we extend the result for general k, achieving $O(k^{O(1/\varepsilon)} n^{1+\varepsilon})$ running time for any $\varepsilon > 0$.

Along the way, we also prove a new $O(k^{O(1)} n \operatorname{polylog} n)$ bound on the number of similar copies of P inside Q that have 4 vertices of P in contact with the boundary of Q (assuming general position input), disproving a conjecture by Agarwal, Amenta, and Sharir (1998).

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Polygon containment, convex polygons, translations, rotations

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.34

Related Version *Full Version*: https://arxiv.org/abs/2403.13292

Funding *Timothy M. Chan*: Work supported by NSF Grant CCF-2224271.

1 Introduction

Polygon containment problems have been studied since the early years of computational geometry [1, 2, 6, 9, 12, 13, 14, 16, 17, 20, 22, 24, 25, 26, 27, 30]. In this paper, we focus on two of the most fundamental versions of the problem for convex polygons:

▶ **Problem 1.** *Given a convex k-gon P and a convex n-gon Q in $\mathbb{R}^2$, (i) find a congruent copy of P inside Q (if it exists); or more generally, (ii) find the largest similar copy of P inside Q.*

In a congruent copy, we allow translation and rotation; in a similar copy, we allow translation, rotation, and scaling. Rotation is what makes the problem challenging, as the corresponding problem without rotation can be solved in linear time by a simple reduction to linear programming in 3 variables [30].

There were 3 key prior papers on this problem:

1. In 1983, Chazelle [12] initiated the study of polygon containment problems and presented an $O(kn^2)$-time algorithm specifically for Problem 1(i). In particular, an entire section of his paper was devoted to an $O(n^2)$-time algorithm just for the $k = 3$ (triangle) case.
2. Sharir and Toledo [30] (preliminary version in SoCG'91) applied parametric search [23] to reduce various versions of "extremal" polygon containment problems (about finding largest copies) to their corresponding decision problems. In particular, they described an $O(n^2 \log^2 n)$-time algorithm for Problem 1(ii) in the $k = 3$ case.

3. In 1998, Agarwal, Amenta, and Sharir [1] studied Problem 1(ii) and obtained an $O(kn^2 \log n)$-time algorithm. Their approach is to explore the entire solution space. More precisely, consider a standard 4-parameter representation of similarity transformations [7]: given $(s,t,u,v) \in \mathbb{R}^4$, let $\varphi_{s,t,u,v} : \mathbb{R}^2 \to \mathbb{R}^2$ be the similarity transformation $(x,y) \mapsto (sx - ty + u, tx + sy + v)$, which has scaling factor $\sqrt{s^2+t^2}$. The region

$$\mathcal{P} = \{(s,t,u,v) \in \mathbb{R}^4 : \ \varphi_{s,t,u,v}(p) \in Q \text{ for all vertexes } p \text{ of } P\}$$

describes all feasible solutions and is an intersection of $O(kn)$ halfspaces in $\mathbb{R}^4$ (since $\varphi_{s,t,u,v}(p)$ is a linear function in the 4 variables s,t,u,v for any fixed point p). The problem is to find a point in $\mathcal{P}$ maximizing the convex function s^2+t^2 (the optimum must be located at a vertex). By standard results, a 4-polytope with $O(kn)$ facets has $O(k^2n^2)$ combinatorial complexity (and can be constructed in $O(k^2n^2)$ time) [28]. Agarwal, Amenta, and Sharir improved the combinatorial bound to $O(kn^2)$ for this particular polytope $\mathcal{P}$, enabling them to derive an algorithm with a similar time bound.

Notice that all these previous algorithms have $\Omega(n^2)$ time complexity, even in the triangle ($k=3$) case. (Other quadratic algorithms for $k=3$ have been found, e.g., mostly recently by Lee, Eom, and Ahn [22].) To explain why, Chazelle [12] mentioned that there are input convex polygons Q for which the number of different "stable solutions" is $\Omega(n^2)$. (Other authors made similar observations [22].) More generally, Agarwal, Amenta, and Sharir [1] exhibited a construction of input convex polygons P and Q for which the polytope $\mathcal{P}$ has complexity $\Omega(kn^2)$, matching their combinatorial upper bound.

Although such combinatorial lower bound results do not technically rule out the possibility of faster algorithms that find an optimal solution without generating the entire solution space, they indicate that quadratic complexity is a natural barrier. In general, no techniques are known to maximize a convex function in an intersection of halfspaces in constant dimensions with worst-case time better than constructing the entire halfspace intersection.

What motivates us to revisit this topic is the similarity of the $k=3$ problem to the well-known *3SUM problem* (in the sense that the main case of the triangle problem is about finding a *triple* of vertices/edges of Q in contact with the triangle P). Our initial thought is to apply the exciting recent advances for 3SUM and related problems [5, 8, 11, 18, 19] to design decision trees with subquadratic height. This would potentially lead to slightly subquadratic algorithms with running time of the form $n^2/\operatorname{polylog} n$.

Although we believe this line of attack can indeed be applied to Problem 1 in the $k=3$ case, the improvement in the time complexity would be tiny, and generalization to $k>3$ is unclear. Furthermore, the usage of such heavy machinery might seem premature and unjustified since the $k=3$ problem has not been shown to be 3SUM-hard. Barequet and Har-Peled [9] proved that Problem 1(i) for convex polygons is 3SUM-hard when $k=n$ and so has a near-quadratic conditional lower bound, but for $k=n$, the current upper bound is cubic. More recently, Künnemann and Nusser [20] have obtained conditional lower bounds for a number of other polygon containment problems, but not in the convex cases.

New results. In this paper, we not only truly break the quadratic barrier but also discover a near-linear, $O(n \operatorname{polylog} n)$-time algorithm for Problem 1(ii) in the $k=3$ case! This represents a substantial improvement over the previous algorithms from multiple decades earlier, and directly addresses an open problem posed by Agarwal, Amenta, and Sharir [1] asking for an algorithm faster than $\Theta(kn^2)$ time. (We cannot think of too many classical 2D problems in computational geometry of comparable stature where quadratic/superquadratic time complexity is reduced to near-linear in a single swoop after a long gap. The closest analog is perhaps Sharir's breakthrough $O(n \operatorname{polylog} n)$-time algorithm for the 2D Euclidean 2-center problem [29] that improved a string of previous $O(n^2 \operatorname{polylog} n)$-time algorithms.)

Furthermore, we generalize our approach and obtain an $O(n^{1+\varepsilon})$-time algorithm for Problem 1(ii) for any constant $k > 3$, where $\varepsilon > 0$ is an arbitrarily small constant. For non-constant $k \le n$, the time bound is $O(k^{O(1/\varepsilon)} n^{1+\varepsilon})$ for any choice of (possibly non-constant) $\varepsilon > 0$. (By choosing $\varepsilon = \sqrt{\log k / \log n}$, the bound can be rewritten as $n 2^{O(\sqrt{\log k \log n})}$.) This beats the previous $O(kn^2)$ bound for all $k < n^\alpha$ for some concrete constant $\alpha > 0$.

New approach. Although Problem 1(ii) reduces to Problem 1(i) by parametric search [23, 30] (if one does not mind extra logarithmic factors), we actually find it more convenient to solve Problem 1(ii) directly (i.e., find the largest copy). The optimal solution must belong to one of the following cases, as observed in previous works (by simple direct arguments, or by recalling that the optimum corresponds to a vertex of the 4-polytope $\mathcal{P}$):

- **2-Contact (i.e., 2-Anchor) Case:** 2 distinct vertices of P are in contact with ∂Q, both of which are at 2 vertices of Q. These 2 vertices of P are called the 2 "anchor" vertices.
- **3-Contact (i.e., 1-Anchor) Case:** 3 distinct vertices of P are in contact with ∂Q, one at a vertex of Q and the other two on edges of Q. The vertex of P placed at a vertex of Q is referred to as the "anchor" vertex.
- **4-Contact (i.e., No-Anchor) Case:** 4 distinct vertices of P are in contact with ∂Q, all on edges of Q.

For $k = 3$, the main case is the 3-contact case, since it turns out that the 2-contact case can be solved in a similar way (and the 4-contact case, of course, does not arise). The overall strategy is to divide into sub-problems involving different "arcs" (i.e., contiguous pieces) of ∂Q. Our key observation is that under certain conditions about the slopes/angles of the input arcs, all 3-contact feasible solutions may be covered by just a linear number of pairs of sub-edges, due to *monotonicity* arguments – this is despite the fact that the total number of 3-contact solutions may be quadratic. In such scenarios, we can search for the best solution by using standard geometric data structuring techniques (concerning intersections of ellipses, as it turns out). A simple binary divide-and-conquer reduces to instances where such conditions are met, resulting in an $O(n \operatorname{polylog} n)$-time algorithm.

For $k > 3$, extending the 3-contact algorithm requires more technical effort (and a slightly increased running time), but what appears even more challenging is the 4-contact case. The lack of anchor vertices seems to make everything more complicated (including the needed geometric data structures). However, with a different strategy, we show surprisingly that the 4-contact case is easier in the sense that the total number of 4-contact feasible solutions is actually near-linear in n, namely, $O(k^4 n \operatorname{polylog} n)$ (assuming general position input). Thus, we can afford to enumerate them all! We prove this combinatorial bound by running our $k = 3$ algorithm on different triples of vertices of P and then piecing information together via further interesting monotonicity arguments.

To see how counterintuitive our near-linear combinatorial bound for 4-contact solutions is, recall that Agarwal, Amenta, and Sharir [1] proved an $\Omega(kn^2)$ lower bound on the size of the solution space. They noted that their construction only lower-bounded the number of 3-contact solutions, and at the end of their paper, they asked for another construction with $\Omega(kn^2)$ 4-contact solutions. Our proof answers their question in the negative.

Preliminaries. The *angle* of a line $\overleftrightarrow{p_1p_2}$, denoted $\theta_{p_1p_2}$, refers to the angle measured counterclockwise (ccw) from the x-axis to $\overleftrightarrow{p_1p_2}$. (Note that $\theta_{p_1p_2} \in [0, \pi)$ and $\theta_{p_2p_1} = \theta_{p_1p_2}$.) An *arc* Γ of a convex polygon Q refers to a contiguous portion of the boundary ∂Q whose supporting lines have angles in an interval of length $< \pi/3$. Let $\Lambda(\Gamma)$ (the *angle range* of

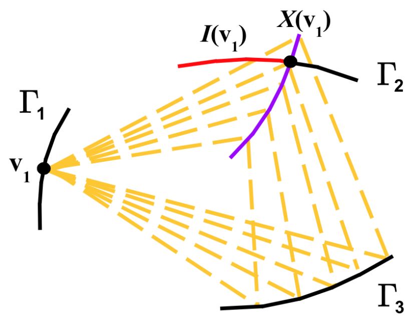

Figure 1 The construction from Lemma 1. Each dashed triangle is a similar copy of $\triangle p_1p_2p_3$, and the purple arc is $f_{v_1}(\Gamma_3)$. (We draw Γ_2 and Γ_3 instead of $\overleftrightarrow{\Gamma_2}$ and $\overleftrightarrow{\Gamma_3}$ for visual clarity.)

Γ) denote the interval containing the angles of all supporting lines of Γ. We allow $\Lambda(\Gamma)$ to wrap around (mod π), so $[a,b]$ indicates $[a,\pi)\cup[0,b]$ if $a>b$. We assume that no polygons contain parallel adjacent edges, as any such edges can be merged.

We assume that all polygon boundaries and their edges/arcs are oriented in ccw order. For each edge e of Q, let $\overleftrightarrow{e}$ denote its extension as an oriented line (with Q on its "left" side). For an arc Γ, let $\overleftrightarrow{\Gamma}$ denote an extension of Γ where the first and last edge are extended to rays (again oriented with Q on its "left" side). We use $\widetilde{O}$ notation to hide polylog n factors.

2 3-Contact (1-Anchor) Case

In this section, we solve the 3-contact case, where there is 1 anchor vertex p_1 of P. We will first present an algorithm for $k=3$ (when P is a triangle), and then we discuss how to generalize it for $k>3$. We will divide into sub-problems operating on different arcs. For $k=3$, the goal is to find a placement where the anchor p_1 is at a vertex on an arc Γ_1, and the two other vertices p_2 and p_3 of P are on edges of arcs Γ_2 and Γ_3.

2.1 An Easy "Disjoint" Case for $k=3$

We begin with an easy lemma to handle the case when the angle ranges for Γ_2 and Γ_3, after suitable rotational shifts, are disjoint.

▶ **Lemma 1.** *Let $\triangle p_1p_2p_3$ be a triangle. Let $\Gamma_1,\Gamma_2,\Gamma_3$ be arcs of a convex n-gon Q, s.t.*

- *$\Lambda(\Gamma_2)+\theta_{p_1p_3}$ and $\Lambda(\Gamma_3)+\theta_{p_1p_2}$ are disjoint (mod π).*

In $\widetilde{O}(|\Gamma_1|)$ time, we can compute, for each vertex v_1 of Γ_1, a point $X(v_1)$ on Γ_2 and a sub-arc $I(v_1)$ of Γ_2, satisfying the following property:[1]

> *For every similarity transformation φ that has $\varphi(p_1)$ being a vertex v_1 of Γ_1 and $\varphi(p_2)$ on Γ_2, we have: (i) $\varphi(p_3)$ is on Γ_3 iff $\varphi(p_2)=X(v_1)$, and (ii) $\varphi(p_3)$ is left of $\overleftrightarrow{\Gamma_3}$ iff $\varphi(p_2)$ is on $I(v_1)$.*

Proof. Let $f_{v_1}(\zeta)$ be the point $\varphi(p_2)$ for the unique similarity transformation φ with $\varphi(p_1)=v_1$ and $\varphi(p_3)=\zeta$. In other words, f_{v_1} is a similarity transformation that keeps v_1 fixed and sends p_3 to p_2, i.e., we rotate around v_1 by an angle $\theta_{p_1p_2}-\theta_{p_1p_3}+\{0,\pm\pi\}$, and scale by

[1] We allow $X(v_1)$ to be undefined and $I(v_1)$ to be empty.

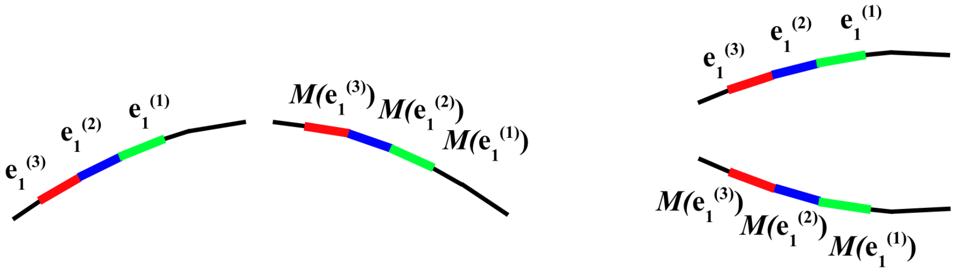

Figure 2 (Left) Monotonically increasing pairing. (Right) Monotonically decreasing pairing.

factor $\|p_1 - p_2\|/\|p_1 - p_3\|$. Thus, $f_{v_1}(\Gamma_3)$ is a similar copy of Γ_3. The supporting lines for $f_{v_1}(\Gamma_3)$ have angles in $\Lambda(\Gamma_3) + \theta_{p_1p_2} - \theta_{p_1p_3} + \{0, \pm\pi\}$, which by the assumption is disjoint from $\Lambda(\Gamma_2)$ (mod π). Thus,[2] $\overleftrightarrow{f_{v_1}(\Gamma_3)}$ and $\overleftrightarrow{\Gamma_2}$ intersect once, at a unique point ν, which we define as $X(v_1)$, and which can be computed by binary search (see Figure 1). We can define $I(v_1)$ to be a prefix or suffix of Γ_2 delimited by $X(v_1)$. ◀

Thus, to solve the 3-contact problem for $k = 3$, we can just examine the unique similarity transformation φ with $\varphi(p_1) = v_1$ and $\varphi(p_2) = X(v_1)$ for each vertex v_1 of Γ_1, in near-linear total time (assuming the disjointness condition is met). It suffices to consider only this single similarity transformation for each $v_1 \in \Gamma_1$, since this is the only possible 3-contact placement.

2.2 A "Double-Disjoint" Case for $k = 3$

Next, we address a different case where the angle ranges for Γ_1 and Γ_2 are disjoint and the angle ranges for Γ_1 and Γ_3 are disjoint, after appropriate rotational shifts. The following lemma reveals a crucial monotonicity phenomenon that we will repeatedly exploit.

To state the lemma, we first introduce some definitions: For two arcs Γ_1 and Γ_2, a *pairing* M between Γ_1 and Γ_2 refers to a subdivision of the (straight) edges of Γ_1 and Γ_2 into sub-edges, together with a bijective mapping between the sub-edges of Γ_1 and the sub-edges of Γ_2. For a sub-edge e_1 of Γ_1, we use $M(e_1)$ to denote e_1's corresponding sub-edge in Γ_2; similarly, for a sub-edge e_2 of Γ_2, we use $M(e_2)$ to denote e_2's corresponding sub-edge in Γ_1. We say that the pairing M is *monotonically increasing (resp. decreasing)* if $M(e_1)$ always advances in ccw (resp. cw) order in Γ_2 as e_1 advances in ccw order in Γ_1 (see Figure 2).

▶ **Lemma 2** (Pairing Lemma). *Let $\triangle p_1p_2p_3$ be a triangle. Let $\Gamma_1, \Gamma_2, \Gamma_3$ be arcs of a convex n-gon Q, such that*

1. *$\Lambda(\Gamma_1) + \theta_{p_2p_3}$ and $\Lambda(\Gamma_2) + \theta_{p_1p_3}$ are disjoint (mod π), and*
2. *$\Lambda(\Gamma_1) + \theta_{p_2p_3}$ and $\Lambda(\Gamma_3) + \theta_{p_1p_2}$ are disjoint (mod π).*

In $\widetilde{O}(n)$ time, we can compute a (monotonically increasing or decreasing) pairing M between Γ_2 and Γ_3 with $O(n)$ sub-edges, satisfying the following property:

> *For every similarity transformation φ that has $\varphi(p_1)$ on Γ_1 and $\varphi(p_2)$ on a sub-edge e_2 of Γ_2, we have: (i) $\varphi(p_3)$ is on Γ_3 iff $\varphi(p_3)$ is on $M(e_2)$; and (ii) $\varphi(p_3)$ is left of $\overleftrightarrow{\Gamma_3}$ iff $\varphi(p_3)$ is left of $\overleftrightarrow{M(e_2)}$.*

[2] This is analogous to the fact that if f and g are functions over $\mathbb{R}$ where the ranges of their derivatives f' and g' are contained in two disjoint closed intervals, then f and g intersect once.

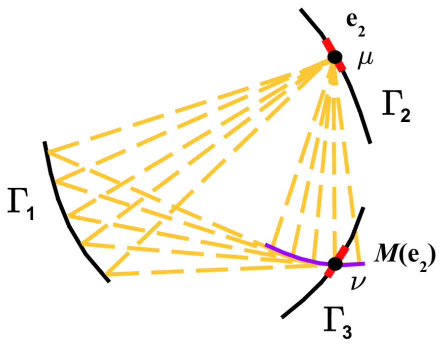

Figure 3 An example of a pairing. Each dashed triangle is a similar copy of $\triangle p_1p_2p_3$.

Proof. We *match* a point μ on $\overleftrightarrow{\Gamma_2}$ with a point ν on $\overleftrightarrow{\Gamma_3}$ iff there exists a similarity transformation φ with $\varphi(p_2) = \mu$, $\varphi(p_1)$ on $\overleftrightarrow{\Gamma_1}$, and $\varphi(p_3) = \nu$.

Observe that a point μ on $\overleftrightarrow{\Gamma_2}$ matches a unique point ν on $\overleftrightarrow{\Gamma_3}$. To see this, let $f_\mu(\zeta)$ be the point $\varphi(p_3)$ for the unique similarity transformation φ with $\varphi(p_2) = \mu$ and $\varphi(p_1) = \zeta$. In other words, f_μ is the similarity transformation that keeps μ fixed and sends p_1 to p_3, i.e., we rotate around μ by an angle $\theta_{p_2p_3} - \theta_{p_2p_1} + \{0, \pm\pi\}$, and scale by factor $\|p_3 - p_2\|/\|p_1 - p_2\|$. Thus, $f_\mu(\Gamma_1)$ is a similar copy of Γ_1. The supporting lines for $f_\mu(\Gamma_1)$ have angles in $\Lambda(\Gamma_1) + \theta_{p_2p_3} - \theta_{p_2p_1} + \{0, \pm\pi\}$, which by assumption 2 is disjoint from $\Lambda(\Gamma_3)$ (mod π). Thus, $f_\mu(\overleftrightarrow{\Gamma_1})$ and $\overleftrightarrow{\Gamma_3}$ intersect once, namely, at the unique point ν. A symmetric argument (swapping subscripts 2 and 3) shows that a point ν on $\overleftrightarrow{\Gamma_3}$ matches a unique point μ on $\overleftrightarrow{\Gamma_2}$, this time, by assumption 1.

Consequently,[3] as μ moves along $\overleftrightarrow{\Gamma_2}$, its matching point ν moves monotonically along $\overleftrightarrow{\Gamma_3}$. We break an edge at the points μ on Γ_2 that match the vertices of $\overleftrightarrow{\Gamma_3}$, which can be found by n binary searches. Similarly, we break an edge at the points ν on Γ_3 that match the vertices of $\overleftrightarrow{\Gamma_3}$, which can be found by n binary searches. As a result, all points μ on a common sub-edge e_2 of $\overleftrightarrow{\Gamma_2}$ are matched with points on a common sub-edge of $\overleftrightarrow{\Gamma_3}$, which we define as $M(e_2)$. For all these points μ, we have $f_\mu(\overleftrightarrow{\Gamma_1})$ intersecting this sub-edge $M(e_2)$ of $\overleftrightarrow{\Gamma_3}$. Also, for all $\zeta \in \Gamma_1$, $f_\mu(\zeta)$ is left of $\overleftrightarrow{\Gamma_3}$ iff $f_\mu(\zeta)$ is left of $\overleftrightarrow{M(e_2)}$ (see Figure 3). ◀

In the above, we did not claim a monotone pairing between Γ_2 and Γ_1, nor between Γ_1 and Γ_3. Otherwise, we would get a linear upper bound on the number of 3-contact solutions in this case, which by our subsequent divide-and-conquer algorithm would yield an $O(n \log n)$ bound on the number of 3-contact solutions in general for $k = 3$, contradicting the known quadratic lower bound [1, 22]! This contradiction does not arise since in the worst case, each matched pair from Γ_2 and Γ_3 could admit legal 3-contact placements with *every* vertex of Γ_1.

With the Pairing Lemma at hand, we can efficiently solve the problem when the two disjointness conditions are met. Specifically, we set up a range searching sub-problem between the $O(n)$ pairs of sub-edges in Γ_2 and Γ_3 (the "data set"), and the $O(n)$ vertices of Γ_1 (the "query points"). This range searching sub-problem turns out to be near-linear-time solvable:

▶ **Lemma 3.** *Let $\triangle p_1p_2p_3$ be a triangle. Let $\Gamma_1, \Gamma_2, \Gamma_3$ be arcs of a convex n-gon Q, s.t.*
1. *$\Lambda(\Gamma_1) + \theta_{p_2p_3}$ and $\Lambda(\Gamma_2) + \theta_{p_1p_3}$ are disjoint (mod π), and*
2. *$\Lambda(\Gamma_1) + \theta_{p_2p_3}$ and $\Lambda(\Gamma_3) + \theta_{p_1p_2}$ are disjoint (mod π).*

[3] This is analogous to the fact that a continuous bijective function over $\mathbb{R}$ must be monotone.

In $\widetilde{O}(n)$ time, we can find a similarity transformation φ, maximizing the scaling factor, such that $\varphi(p_1)$ is a vertex of Γ_1, $\varphi(p_2)$ is on Γ_2, and $\varphi(p_3)$ is on Γ_3.

Proof. Given $(s,t,u,v) \in \mathbb{R}^4$, let $\varphi_{s,t,u,v} : \mathbb{R}^2 \to \mathbb{R}^2$ be the similarity transformation $(x,y) \mapsto (sx - ty + u, tx + sy + v)$ (which has scaling factor $\sqrt{s^2+t^2}$).

Apply Lemma 2 to get a pairing M between Γ_2 and Γ_3. For each sub-edge e_2 of Γ_2, define

$$\begin{aligned}
\mathcal{P}(e_2) &= \{(s,t,u,v) \in \mathbb{R}^4 : \quad \varphi_{s,t,u,v}(p_2) \text{ is on } e_2, \text{ and } \varphi_{s,t,u,v}(p_3) \text{ is on } M(e_2)\} \\
R_\rho(e_2) &= \{\varphi_{s,t,u,v}(p_1) : \quad (s,t,u,v) \in \mathcal{P}(e_2) \text{ and } s^2+t^2 \geq \rho^2\}.
\end{aligned}$$

Observe that $\mathcal{P}(e_2)$ is a 2-dimensional convex polygon in $\mathbb{R}^4$ with $O(1)$ edges (since $\varphi_{s,t,u,v}(p_2)$ and $\varphi_{s,t,u,v}(p_3)$ are linear in the variables s,t,u,v, and the 2 point-on-line-segment conditions yield 2 linear equality constraints and 4 linear inequality constraints in these 4 variables). Furthermore, $R_\rho(e_2)$ is a region in $\mathbb{R}^2$ which is the intersection of a convex $O(1)$-gon with the exterior of an ellipse (since $(s,t,u,v) \mapsto \varphi_{s,t,u,v}(p_1)$ is a linear projection from $\mathbb{R}^4$ to $\mathbb{R}^2$, and the projection of a 2-dimensional slice of the cylinder $\{(s,t,u,v) : s^2+t^2 = \rho^2\}$ is an ellipse).

The decision problem (deciding whether the maximum scaling factor is at least a given value ρ) reduces to finding a pair of vertex v_1 of Γ_1 and sub-edge e_2 of Γ_2, such that $v_1 \in R_\rho(e_2)$. To this end, we will build a data structure to store the $O(n)$ regions $R_\rho(e_2)$ over all e_2 so that we can quickly decide whether the query point v_1 stabs (i.e., is contained in) some region $R_\rho(e_2)$.

We use standard techniques in geometric data structures. First, we consider the *range stabbing* problem for exteriors of ellipses: build a data structure for a set of $O(n)$ ellipses in $\mathbb{R}^2$, so that we can quickly decide whether a query point stabs the exterior of some ellipse, i.e., whether a query point is outside the intersection of the interiors of the ellipses. The intersection of the interiors of $O(n)$ ellipses (which is a single cell in the arrangement) has almost linear combinatorial complexity by standard results on Davenport-Schinzel sequences [4], and can be constructed in $\widetilde{O}(n)$ time, e.g., by divide-and-conquer. Thus, this problem can be solved with $\widetilde{O}(n)$ preprocessing time and $\widetilde{O}(1)$ query time.

Next, we consider range stabbing for our regions $R_\rho(e_2)$. As each region is the intersection of a convex $O(1)$-gon with the exterior of an ellipse, we can use standard *multi-level* data structuring techniques [3]to handle the extra $O(1)$ halfplane constraints. Generally, halfplane range searching cannot be solved with near-linear preprocessing time and polylogarithmic query time. But in our application, all query points v_1 lie on a convex chain Γ_1. The constraint that such a query point v_1 lies inside a halfplane is equivalent to the condition that v_1 lies inside one of $O(1)$ 1D intervals, assuming that the vertices of Γ_1 are stored in a sorted array. We can therefore use 1D range trees [3, 21, 28] to handle the halfplane constraints, with only a logarithmic factor increase in the preprocessing and query time.

The optimization problem reduces to the decision problem by a standard application of parametric search [23]. (The application requires a parallelization of the decision algorithm: the preprocessing part, namely, the construction of the intersection of interiors of ellipses, is straightforwardly parallelizable by divide-and-conquer; the $O(n)$ queries can trivially be answered in parallel.) Parametric search increases the running time by a polylogarithmic factor. Alternatively, we can apply Chan's randomized optimization technique [10], which avoids extra factors. (The application here is straightforward, since the problem can be viewed as a generalized "closest-pair-type" problem [10] between two sets of objects.) ◀

2.3 Simple Divide-and-Conquer Algorithm for $k = 3$

We now have all the ingredients needed to put together a simple recursive algorithm to solve the $k = 3$ problem in the 3-contact case:

▶ **Theorem 4.** *Given a triangle P and a convex n-gon Q, in $\widetilde{O}(n)$ time, we can find the largest similar copy of P contained in Q that has 1 vertex of P at a vertex of Q and the 2 other vertices of P on edges of Q.*

Proof. Let $P = \triangle p_1p_2p_3$. Arbitrarily divide ∂Q into $O(1)$ arcs, and let $\Gamma_1, \Gamma_2, \Gamma_3$ be 3 such arcs (allowing duplicates). We will try all $O(1)$ choices of $\Gamma_1, \Gamma_2, \Gamma_3$.

Let S be an interval. Let $\Gamma_1(S)$ (resp. $\Gamma_2(S)$ and $\Gamma_3(S)$) be the sub-arc of Γ_1 (resp. Γ_2 and Γ_3) consisting of all edges whose supporting lines have angles in $S - \theta_{p_2p_3}$ (resp. $S - \theta_{p_1p_3}$ and $S - \theta_{p_1p_2}$) (mod π). We will recursively solve the following problem: find a similarity transformation φ, maximizing the scaling factor, such that $\varphi(p_1)$ is a vertex v_1 of $\Gamma_1(S)$, $\varphi(p_2)$ is on $\Gamma_2(S)$, and $\varphi(p_3)$ is on $\Gamma_3(S)$.

As a first step, we remove edges not participating in $\Gamma_1(S), \Gamma_2(S), \Gamma_3(S)$, so that the number of edges in Q is reduced to $m(S) := |\Gamma_1(S)| + |\Gamma_2(S)| + |\Gamma_3(S)|$. Partition S into sub-intervals S^- and S^+ (as shown in Figure 4) so that $m(S^-), m(S^+) = m(S)/2 \pm O(1)$. We try various possibilities and take the best solution found:

- Case 1: $\varphi(p_1)$ is on $\Gamma_1(S^-)$, $\varphi(p_2)$ is on $\Gamma_2(S^-)$, and $\varphi(p_3)$ is on $\Gamma_3(S^-)$. We can recursively solve the problem for S^-.
- Case 2: $\varphi(p_1)$ is on $\Gamma_1(S^+)$, $\varphi(p_2)$ is on $\Gamma_2(S^+)$, and $\varphi(p_3)$ is on $\Gamma_3(S^+)$. We can recursively solve the problem for S^+.
- Case 3: $\varphi(p_1)$ is on $\Gamma_1(S^-)$, $\varphi(p_2)$ is on $\Gamma_2(S^+)$, and $\varphi(p_3)$ is on $\Gamma_3(S^+)$. Since $\Lambda(\Gamma_1(S^-)) + \theta_{p_2p_3} \subseteq S^-$ and $\Lambda(\Gamma_2(S^+)) + \theta_{p_1p_3} \subseteq S^+$ are disjoint (mod π), and $\Lambda(\Gamma_1(S^-)) + \theta_{p_2p_3} \subseteq S^-$ and $\Lambda(\Gamma_3(S^+)) + \theta_{p_1p_2} \subseteq S^+$ are disjoint (mod π), we can solve this sub-problem by Lemmas 2–3 in $\widetilde{O}(m(S))$ time.
- Case 4: $\varphi(p_1)$ is on $\Gamma_1(S^-)$, $\varphi(p_2)$ is on $\Gamma_2(S^-)$, and $\varphi(p_3)$ is on $\Gamma_3(S^+)$. Since $\Lambda(\Gamma_2(S^-)) + \theta_{p_1p_3} \subseteq S^-$ and $\Lambda(\Gamma_3(S^+)) + \theta_{p_1p_2} \subseteq S^+$ are disjoint (mod π), we can solve this sub-problem by Lemma 1 in $\widetilde{O}(m(S))$ time. Namely, for each vertex v_1 of $\Gamma_1(S^-)$, we just check the unique similarity transformation φ with $\varphi(p_1) = v_1$ and $\varphi(p_2) = X(v_1)$.

All remaining cases are symmetric to Cases 3 and 4 (swapping subscripts 2 and 3 and/or S^- and S^+).

Letting $m = m(S)$, we obtain the following recurrence for the running time:

$$T(m) \leq 2T(m/2 + O(1)) + \widetilde{O}(m).$$

The recurrence solves to $T(m) = \widetilde{O}(m)$. ◀

It is not difficult to modify the algorithm to also solve the 2-contact (2-anchor) case, as shown in the full paper. This gives a complete $\widetilde{O}(n)$ time algorithm for $k = 3$.

2.4 Generalizing to $k > 3$

With further effort, we can also solve the problem for general k in the 3-contact case. Say p_1 is the anchor vertex, and p_2 and p_3 are the other two vertices of P in contact with Q. The idea is to just apply the Pairing Lemma to the triangles $\triangle p_1p_2p_i$ for all other vertices p_i of P, assuming appropriate disjointness conditions. We need to extend the problem to equip each vertex v_1 of Γ_1 with a sub-arc $I(v_1)$ of Γ_2 which restricts the placement of p_2. The resulting range searching sub-problems can be solved in a manner similar to the triangle case, which we show in the following extension of Lemma 3 (see the full paper for the proof):

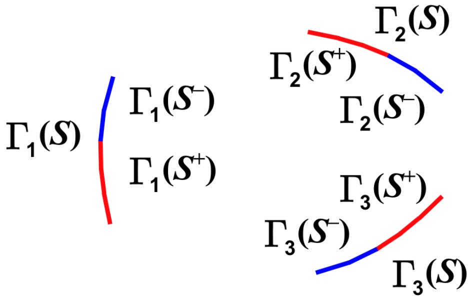

Figure 4 Partitioning each $\Gamma_i(S)$ into $\Gamma_i(S^+)$ and $\Gamma_i(S^-)$, for $i \in \{1, 2, 3\}$.

▶ **Lemma 5.** *Let P be a k-gon with vertices $p_1, \ldots, p_k$ (not necessarily in sorted order). Let $\Gamma_1, \ldots, \Gamma_k$ be arcs of a convex n-gon Q, such that*

1. *for each $i \in \{3, \ldots, k\}$, $\Lambda(\Gamma_1) + \theta_{p_2 p_i}$ and $\Lambda(\Gamma_2) + \theta_{p_1 p_i}$ are disjoint (mod π), and*
2. *for each $i \in \{3, \ldots, k\}$, $\Lambda(\Gamma_1) + \theta_{p_2 p_i}$ and $\Lambda(\Gamma_i) + \theta_{p_1 p_2}$ are disjoint (mod π).*

For each vertex v_1 of Γ_1, we are given a sub-arc $I(v_1)$ of Γ_2. In $\widetilde{O}(k^2 n)$ time, we can find a similarity transformation φ, maximizing the scaling factor, such that $\varphi(p_1)$ is a vertex v_1 of Γ_1, $\varphi(p_2)$ is on $I(v_1) \subseteq \Gamma_2$, $\varphi(p_3)$ is on Γ_3, and for each $i \in \{4, \ldots, k\}$, $\varphi(p_i)$ is left of $\overleftrightarrow{\Gamma_i}$.

We now give a slightly more intricate divide-and-conquer algorithm for general k:

▶ **Theorem 6.** *Given a k-gon P and a convex n-gon Q (where $k \leq n$), we can find the largest similar copy of P contained in Q that has 1 vertex of P at a vertex of Q and the 2 other vertices of P on 2 edges of Q, in $O(k^{O(1/\varepsilon)} n^{1+\varepsilon})$ time for any $\varepsilon > 0$.*

Proof. Suppose the vertices of P are $p_1, \ldots, p_k$ (not necessarily in sorted order). Divide ∂Q into $O(1)$ arcs, and let $\Gamma_1, \Gamma_2, \Gamma_3$ be 3 such arcs (allowing duplicates). We will try all choices for p_1, p_2, p_3 and $\Gamma_1, \Gamma_2, \Gamma_3$; this increases the final running time by a factor of $O(k^3)$.

Let $\Gamma_4, \ldots, \Gamma_k$ be arcs of ∂Q, so that a similarity transformation φ has $\varphi(P)$ inside Q iff $\varphi(p_i)$ is left of $\overleftrightarrow{\Gamma_i}$ for all $i \in \{1, \ldots k\}$. This is w.l.o.g. since we can just make $O(1)$ copies of $p_4, \ldots, p_k$ and associate each copy with an arc of ∂Q, while increasing k by a constant factor. Note that duplicate arcs are allowed, and some of these arcs may even be the same as Γ_1, Γ_2, or Γ_3.

We will describe a recursive algorithm, where the input consists of k arcs $\langle \Gamma_1, \ldots, \Gamma_k \rangle$ together with a sub-arc $I(v_1) \subseteq \Gamma_2$ for every vertex v_1 of Γ_1. (For the initial problem, $I(v_1)$ will be all of Γ_2 for all v_1.) Our algorithm will find a similarity transformation φ, maximizing the scaling factor, such that $\varphi(p_1)$ is a vertex v_1 of Γ_1, $\varphi(p_2)$ is on $I(v_1)$, $\varphi(p_3)$ is on Γ_3, and for all $i \in \{4, \ldots, k\}$, $\varphi(p_i)$ is left of $\overleftrightarrow{\Gamma_i}$.

To this end, let $\mathrm{clip}_i(\Gamma_i, \Gamma_2)$ be the sub-arc of Γ_i consisting of all edges of Γ_i whose supporting lines have angles in $\Lambda(\Gamma_2) + \theta_{p_1 p_i} - \theta_{p_1 p_2}$ (mod π). Let $m(\Gamma_2) = \sum_{i=3}^{k} |\mathrm{clip}_i(\Gamma_i, \Gamma_2)| + |\Gamma_2|$. Partition Γ_1 into r sub-arcs such that each sub-arc γ_1 has $|\Gamma_1|/r \pm O(1)$ edges. Partition Γ_2 into r sub-arcs such that each sub-arc γ_2 has $m(\gamma_2) = m(\Gamma_2)/r \pm O(k)$. (This is possible because for a single edge e_2 of γ_2, $m(e_2) \leq O(k)$.)

By Lemma 1, we first enumerate all similarity transformations φ with $\varphi(p_1)$ on a vertex of Γ_1, $\varphi(p_2)$ on some sub-arc γ_2, and $\varphi(p_3)$ on each of the $O(1)$ contiguous pieces of $\Gamma_3 \setminus \mathrm{clip}_3(\Gamma_3, \gamma_2)$; the disjointness condition in Lemma 1 is satisfied by our definition of clip_i (see Figure 5). This gives $O(|\Gamma_1|)$ transformations to check per γ_2, which requires $\widetilde{O}(|\Gamma_1| \cdot rk)$ time over all γ_2 (checking the feasibility of one transformation takes $O(k \log n)$ time, since we can tell whether any given point is inside Q via binary search [28]).

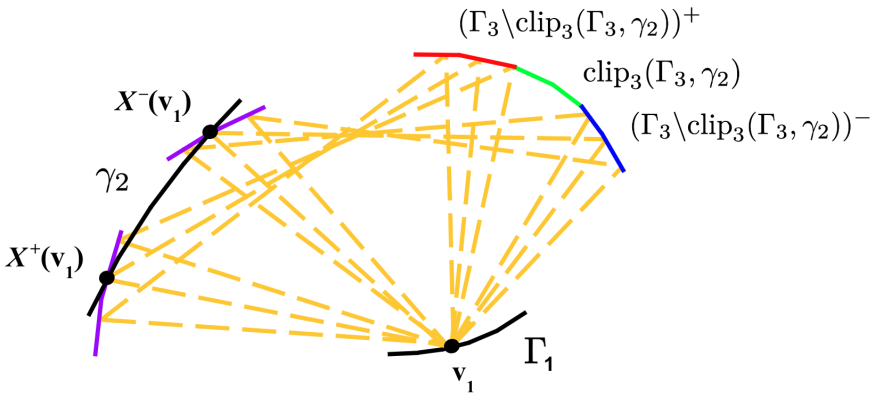

Figure 5 Example where both sub-arcs constituting $\Gamma_3 \setminus \text{clip}_3(\Gamma_3, \gamma_2)$ (denoted $(\Gamma_3 \setminus \text{clip}_3(\Gamma_3, \gamma_2))^+$ and $(\Gamma_3 \setminus \text{clip}_3(\Gamma_3, \gamma_2))^-$) have a similarity transformation that places p_1 at a vertex v_1 of Γ_1, p_2 on γ_2, and p_3 on $\Gamma_3 \setminus \text{clip}_3(\Gamma_3, \gamma_2)$. $X^+(v_1)$ and $X^-(v_2)$ can be found rapidly via Lemma 1 since $(\Lambda(\Gamma_3 \setminus \text{clip}_3(\Gamma_3, \gamma_2)) + \theta_{p_1p_2}) \cap (\Lambda(\gamma_2) + \theta_{p_1p_3}) = \emptyset \pmod{\pi}$. Dashed triangles are similar to $\triangle p_1p_2p_3$.

It remains to search for transformations φ such that $\varphi(p_1)$ is on a vertex of some sub-arc γ_1, $\varphi(p_2)$ is on some sub-arc γ_2, and $\varphi(p_3)$ is on $\text{clip}_3(\Gamma_3, \gamma_2)$. There are two possibilities:

- Case 1: For each $i \in \{3, \ldots, k\}$, $\Lambda(\gamma_1) + \theta_{p_2p_i}$ and $\Lambda(\gamma_2) + \theta_{p_1p_i}$ are disjoint (mod π). For each $i \in \{3, \ldots, k\}$, we apply Lemma 1 to $(\gamma_1, \gamma_2, \gamma')$ for each of the $O(1)$ contiguous pieces γ' of $\Gamma_i \setminus \text{clip}_i(\Gamma_i, \gamma_2)$. For each vertex v_1 of γ_1, let $I'(v_1)$ be the intersection of all sub-arcs $I(v_1)$ produced during these applications of Lemma 1. We now use Lemma 5 on $\langle \gamma_1, \gamma_2, \text{clip}_3(\Gamma_3, \gamma_2), \ldots, \text{clip}_k(\Gamma_k, \gamma_2)\rangle$. The sub-arc we pass to Lemma 5 for each vertex v_1 of γ_1 is $I(v_1) \cap I'(v_1)$. The total time for all instances of this case is $\widetilde{O}(r^2 \cdot k^2(|\Gamma_1|/r + m(\Gamma_2)/r + k))$ (since we can operate on a truncated version of Q consisting of just the arcs/sub-arcs specified).
- Case 2: For some $i \in \{3, \ldots, k\}$, $\Lambda(\gamma_1) + \theta_{p_2p_i}$ and $\Lambda(\gamma_2) + \theta_{p_1p_i}$ intersect (mod π). Here, we recursively solve the problem for $\langle \gamma_1, \gamma_2, \text{clip}_3(\Gamma_3, \gamma_2), \ldots, \text{clip}_k(\Gamma_k, \gamma_2)\rangle$. The sub-arc we pass to the recursive call for each vertex v_1 of γ_1 is $I(v_1) \cap I'(v_1)$, where $I'(v_1)$ is defined as in Case 1. There are $O(r)$ pairs (γ_1, γ_2) satisfying this condition per i, since when we overlay two subdivisions of $\mathbb{R}$ into $O(r)$ intervals, the number of intersecting pairs of intervals is $O(r)$. Thus, the total number of recursive calls for this case is $O(kr)$. The time to produce the sub-problems is subsumed by the time bound in Case 1.

We take the best solution found in all cases. Letting $\widehat{m} = |\Gamma_1| + m(\Gamma_2)$, we obtain the following recurrence for the running time:

$$T(\widehat{m}) \;\le\; O(kr)\,T(\widehat{m}/r + O(k)) + \widetilde{O}(k^2r^2(\widehat{m}/r + k)).$$

As base case, if $\widehat{m} \le kr$, we use the naive bound $T(\widehat{m}) = O(k^2\widehat{m}^2)$ by constructing the space of all feasible placements [1]. The recurrence solves to $T(\widehat{m}) \le O(k)^{\log_r \widehat{m}} \cdot (kr)^{O(1)}\widehat{m}$. Choosing $r = \widehat{m}^{\varepsilon}$ yields $T(\widehat{m}) \le k^{O(1/\varepsilon)}\widehat{m}^{1+O(\varepsilon)}$. (We can adjust ε by a constant factor.) ◀

Note that our earlier divide-and-conquer approach in Theorem 4 (which yielded a slightly better $O(n \operatorname{polylog} n)$ running time) does not work here. This is because we need to ensure disjointness conditions for multiple triangles $\triangle p_1p_2p_i$ with different rotational shifts, meaning that we cannot use one common interval S to represent the k arcs in a sub-problem.

It is not difficult to modify the algorithm to solve the 2-contact (2-anchor) case, as shown in the full paper.

Extension to the 4-contact case is more challenging, however. Without an anchor vertex, Lemma 1 is no longer applicable, so we cannot clip arcs as in Theorem 6's proof. With some care, we could still use the Pairing Lemma to solve the problem, but we would need to pair some arcs Γ_i with Γ_2 and some arcs Γ_i with another arc such as Γ_1. As a result, the range searching sub-problems become more complex, and the running time would be much larger (though subquadratic). In the next section, we suggest a better, more elegant way to solve the 4-contact case in near-linear time, without needing range searching at all!

3 4-Contact (No-Anchor) Case

To solve the 4-contact case, we will actually show that the number of solutions (not necessarily optimal nor locally optimal) is actually near-linear in n, assuming general position input.[4] This allows us to focus on the problem of enumerating all 4-contact placements (as we can check their feasibility rapidly). For the enumeration problem, we can immediately reduce the general k case to the $k = 4$ case.

3.1 Covering All 3-Contact Solutions by Pairs and Triples for $k = 3$

To solve the enumeration problem for $k = 4$, we will actually revisit the $k = 3$ case. Although the number of 3-contact solutions may be quadratic in the worst case, we observe that our divide-and-conquer algorithm from Theorem 4 can generate a near-linear number of pairs that "cover" all 3-contact solutions. To be precise, we make the following definitions: We say that (q_1, q_2, q_3) is *covered* by a list L of triples of edges if q_1 is on e_1, q_2 is on e_2, and q_3 is on e_3 for some $(e_1, e_2, e_3) \in L$. We say that (q_1, q_2) is *covered* by a pairing M of sub-edges if q_1 is on e_1 and q_2 is on $M(e_1)$ for some sub-edge e_1.

We begin with a variant of the Pairing Lemma that guarantees monotonically *increasing* pairings, which will be crucial later (see the full paper for the proof):

▶ **Lemma 7** (Modified Pairing Lemma). *Let $\triangle p_1p_2p_3$ be a triangle. Let $\Gamma_1, \Gamma_2, \Gamma_3$ be arcs of a convex n-gon Q, such that*

1. *$\Lambda(\Gamma_1) + \theta_{p_2p_3}$ and $\Lambda(\Gamma_2) + \theta_{p_1p_3}$ are disjoint (mod π), and*
2. *$\Lambda(\Gamma_1) + \theta_{p_2p_3}$ and $\Lambda(\Gamma_3) + \theta_{p_1p_2}$ are disjoint (mod π).*

In $\widetilde{O}(n)$ time, we can compute a monotonically increasing *pairing M between Γ_2 and Γ_3 with $O(n)$ sub-edges, or a list L of $O(n)$ triples of edges, satisfying the following property:*

> *For every similarity transformation φ that has $\varphi(p_1)$ on Γ_1 and $\varphi(p_2)$ on Γ_2 and $\varphi(p_3)$ on Γ_3, we have $(\varphi(p_2), \varphi(p_3))$ covered by M or $(\varphi(p_1), \varphi(p_2), \varphi(p_3))$ covered by L.*

By adapting our $k = 3$ algorithm, we can cover all 3-contact solutions by $O(\log n)$ monotonically increasing pairings, together with an extra set of $O(n \log n)$ triples:

▶ **Theorem 8.** *Let $\triangle p_1p_2p_3$ be a triangle. Let $\Gamma_1, \Gamma_2, \Gamma_3$ be arcs of a convex n-gon Q. In $\widetilde{O}(n)$ time, we can compute a collection $\mathcal{M}$ of $O(\log n)$ monotonically* increasing *pairings between Γ_1 and Γ_2, between Γ_2 and Γ_3, and between Γ_1 and Γ_3, each with $O(n)$ sub-edges, and a list L of $O(n \log n)$ triples of edges, satisfying the following property:*

[4] In degenerate scenarios, e.g., when P and Q are squares, there could technically be an infinite number of 4-contact placements; in such cases, we may apply small perturbations, or instead count the number of distinct quadruples of edges of Q corresponding to such placements.

For every similarity transformation φ that has $\varphi(p_1)$ on Γ_1 and $\varphi(p_2)$ on Γ_2 and $\varphi(p_3)$ on Γ_3, we have $(\varphi(p_1), \varphi(p_2))$ or $(\varphi(p_2), \varphi(p_3))$ or $(\varphi(p_1), \varphi(p_3))$ covered by some pairing in $\mathcal{M}$, or $(\varphi(p_1), \varphi(p_2), \varphi(p_3))$ covered by L.

Proof. We modify the divide-and-conquer algorithm in the proof of Theorem 4. Let S be an interval. Define $\Gamma_1(S), \Gamma_2(S), \Gamma_3(S)$ as before. We will recursively solve the problem for $\Gamma_1(S), \Gamma_2(S), \Gamma_3(S)$.

As a first step, we remove edges not participating in $\Gamma_1(S), \Gamma_2(S), \Gamma_3(S)$, so that the number of edges in Q reduced to $m(S) := |\Gamma_1(S)| + |\Gamma_2(S)| + |\Gamma_3(S)|$. Divide S into two disjoint sub-intervals S^- and S^+ so that $m(S^-), m(S^+) = m(S)/2 \pm O(1)$. We consider various possibilities:

- Case 1: $\varphi(p_1)$ is on $\Gamma_1(S^-)$, $\varphi(p_2)$ is on $\Gamma_2(S^-)$, and $\varphi(p_3)$ is on $\Gamma_3(S^-)$. We can recursively solve the problem for S^-.
- Case 2: $\varphi(p_1)$ is on $\Gamma_1(S^+)$, $\varphi(p_2)$ is on $\Gamma_2(S^+)$, and $\varphi(p_3)$ is on $\Gamma_3(S^+)$. We can recursively solve the problem for S^+.
- Case 3: $\varphi(p_1)$ is on $\Gamma_1(S^-)$, $\varphi(p_2)$ is on $\Gamma_2(S^+)$, and $\varphi(p_3)$ is on $\Gamma_3(S^+)$. Since $\Lambda(\Gamma_1(S^-)) + \theta_{p_2p_3} \subseteq S^-$ and $\Lambda(\Gamma_2(S^+)) + \theta_{p_1p_3} \subseteq S^+$ are disjoint, and $\Lambda(\Gamma_1(S^-)) + \theta_{p_2p_3} \subseteq S^-$ and $\Lambda(\Gamma_3(S^+)) + \theta_{p_1p_2} \subseteq S^+$ are disjoint (mod π), we can solve the problem by Lemma 7 in $\widetilde{O}(m(S))$ time.

All remaining cases are symmetric to Case 3. (The previous Case 4 is now symmetric to Case 3, since p_1 is no longer treated as a special anchor vertex.)

A pairing between $\Gamma_1(S^-)$ and $\Gamma_2(S^-)$ and a pairing between $\Gamma_1(S^+)$ and $\Gamma_2(S^+)$ produced by the recursive calls in Cases 1 and 2 can be joined into one pairing while remaining monotonically increasing, since $\Gamma_1(S^-)$ precedes $\Gamma_1(S^+)$ and $\Gamma_2(S^-)$ precedes $\Gamma_2(S^+)$ in ccw order. We can join the pairings for Γ_2, Γ_3 and Γ_1, Γ_3 similarly. (And we can trivially union the lists of triples together.)

This yields a total of $O(\log n)$ pairings each with $O(n)$ sub-edges, plus an extra list of $O(n \log n)$ triples. ◀

3.2 Enumerating All 4-Contact Solutions for $k = 4$

To solve the enumeration problem for $k = 4$, we claim that we do not need any further ingredients! We can just run our $k = 3$ algorithm for each of the 4 triangles from the input 4-gon and then piece the outputs together in a careful way.

▶ **Lemma 9.** *Let $p_1p_2p_3p_4$ be a 4-gon and Q be a convex n-gon in general position. Given 3 edges e_1, e_2, e_3 of Q, there are only $O(1)$ similarity transformations φ with $\varphi(p_1)$ on e_1, $\varphi(p_2)$ on e_2, $\varphi(p_3)$ on e_3, and $\varphi(p_4)$ on ∂Q, and they can be computed in $O(\log n)$ time.*

Proof. See the full paper. ◀

▶ **Theorem 10.** *Let P be a 4-gon with vertices p_1, p_2, p_3, p_4 and Q be a convex n-gon in general position. There are at most $O(n \log^2 n)$ similarity transformations φ such that $\varphi(p_1), \varphi(p_2), \varphi(p_3), \varphi(p_4)$ are on ∂Q, and they can be enumerated in $\widetilde{O}(n)$ time.*

Proof. Divide ∂Q into $O(1)$ arcs, and let $\Gamma_1, \dots, \Gamma_4$ be 4 such arcs of Q. (We will try all $O(1)$ choices for $\Gamma_1, \dots, \Gamma_4$.) We apply Theorem 8 to the 4 triangles $\triangle p_1p_2p_3$, $\triangle p_1p_2p_4$, $\triangle p_1p_3p_4$, and $\triangle p_2p_3p_4$, to get a combined collection $\mathcal{M}$ of $O(\log n)$ monotonically increasing pairings and a combined list L of triples.

We consider various possibilities (we will try them all and return the union of the outputs). If $(\varphi(p_1), \varphi(p_2), \varphi(p_3))$, $(\varphi(p_1), \varphi(p_2), \varphi(p_4))$, $(\varphi(p_1), \varphi(p_3), \varphi(p_4))$, or $(\varphi(p_2), \varphi(p_3), \varphi(p_4))$ is covered by L, we can examine each of the $O(n \log n)$ triples in L, and use Lemma 9 to generate $O(1)$ transformations φ per triple, in $\widetilde{O}(n)$ time.

Otherwise, define a small graph G_φ with vertices $\{1, 2, 3, 4\}$, where ij is an edge iff $(\varphi(p_i), \varphi(p_j))$ is covered by a pairing in $\mathcal{M}$. We know that each of triple of vertices contains an edge in G_φ. It is easy to see (from a short case analysis) that G_φ must have ≥ 2 edges.

- Case 1: G_φ contains 2 adjacent edges, w.l.o.g., 12 and 23. Then $(\varphi(p_1), \varphi(p_2))$ is covered by a pairing $M_{12} \in \mathcal{M}$ between Γ_1 and Γ_2, and $(\varphi(p_2), \varphi(p_3))$ is covered by a pairing $M_{23} \in \mathcal{M}$ between Γ_2 and Γ_3. We overlay the 2 subdivisions along Γ_2. We examine the triple $(M_{12}(e_2), e_2, M_{23}(e_2))$ for each sub-edge e_2 of Γ_2, and use Lemma 9 to generate $O(1)$ transformations φ per triple, in $\widetilde{O}(n)$ time. The total number of resulting triples over all $O(\log^2 n)$ choices of M_{12} and M_{23} is $O(n \log^2 n)$, giving $O(n \log^2 n)$ transformations.
- Case 2: G_φ contains 2 independent edges, w.l.o.g., 12 and 34. Then $(\varphi(p_1), \varphi(p_2))$ is covered by a pairing $M_{12} \in \mathcal{M}$ between Γ_1 and Γ_2, and $(\varphi(p_3), \varphi(p_4))$ is covered by a pairing $M_{34} \in \mathcal{M}$ between Γ_3 and Γ_4.

  For 2 sub-edges e and e', define the angle interval $\Theta(e, e') = \{\theta_{qq'} : q \in e,\ q' \in e'\}$. It suffices to enumerate quadruples $(e_1, M_{12}(e_1), e_3, M_{34}(e_3))$ over all sub-edges e_1 of Γ_1 and all sub-edges e_3 of Γ_3, under the restriction that $\Theta(e_1, M_{12}(e_1)) - \theta_{p_1p_2}$ intersects $\Theta(e_3, M_{34}(e_3)) - \theta_{p_3p_4} \pmod{\pi}$. See Figure 6 for an example quadruple.

  Observe that because M_{12} is monotonically increasing, the angle intervals $\Theta(e_1, M_{12}(e_1))$ are disjoint[5] and move monotonically as e_1 moves in ccw order. Similarly, because M_{34} is monotonically increasing, the angle intervals $\Theta(e_3, M_{34}(e_3))$ are disjoint and move monotonically as e_3 moves in ccw order. By overlaying the 2 sets of $O(n)$ intervals $\{\Theta(e_1, M_{12}(e_1)) - \theta_{p_1p_2} :$ sub-edge e_1 of $\Gamma_1\}$ and $\{\Theta(e_3, M_{34}(e_3)) - \theta_{p_3p_4} :$ sub-edge e_3 of $\Gamma_3\}$, we see that there are at most $O(n)$ choices of (e_1, e_3) such that $\Theta(e_1, M_{12}(e_1)) - \theta_{p_1p_2}$ intersects $\Theta(e_3, M_{34}(e_3)) - \theta_{p_3p_4} \pmod{\pi}$, and they can be enumerated in $O(n)$ time. The total number of quadruples over all $O(\log^2 n)$ choices of M_{12} and M_{34} is $O(n \log^2 n)$, giving $O(n \log^2 n)$ transformations. ◀

Interestingly, Case 2 exploits a different phenomenon than in our earlier proofs: besides monotone pairings of sub-edges, we have monotone "pairings of pairs" of sub-edges.

3.3 Generalizing to $k > 4$

Finally, an algorithm for the 4-contact case for general k immediately follows:

▶ **Corollary 11.** *Given a k-gon P and a convex n-gon Q in general position, there are at most $O(k^4 n \log^2 n)$ similar copies of P contained in Q that have 4 different vertices of P on 4 edges of Q, and they can be enumerated in $\widetilde{O}(k^5 n)$ time.*

Proof. For each of the $O(k^4)$ choices of vertices p_1, p_2, p_3, p_4 of P, we generate the $O(n \log^2 n)$ similarity transformations from Theorem 10 and test the feasibility of each transformation in $O(k \log n)$ time via binary searches [15]. ◀

[5] $\Theta(e_1, M_{12}(e_1))$ and $\Theta(e'_1, M_{12}(e'_1))$ may share a limit point if e_1 and e'_1 are adjacent, but this does not affect the proof.

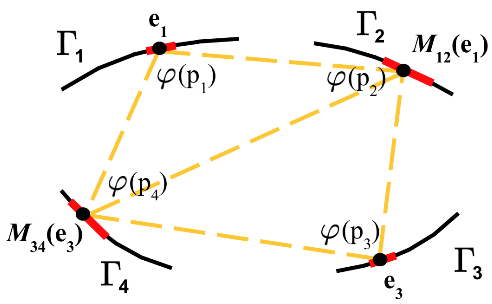

Figure 6 An example quadruple from Case 2 in the proof of Theorem 10. The pairing M_{12} is produced using $\triangle p_1 p_2 p_4$ and the pairing M_{34} is produced using $\triangle p_2 p_3 p_4$.

References

1 Pankaj K. Agarwal, Nina Amenta, and Micha Sharir. Largest Placement of One Convex Polygon Inside Another. *Discret. Comput. Geom.*, 19(1):95–104, 1998. `doi:10.1007/PL00009337`.

2 Pankaj K. Agarwal, Boris Aronov, and Micha Sharir. Motion Planning for a Convex Polygon in a Polygonal Environment. *Discret. Comput. Geom.*, 22(2):201–221, 1999. `doi:10.1007/PL00009455`.

3 Pankaj K. Agarwal and Jeff Erickson. Geometric Range Searching and Its Relatives. *Contemporary Mathematics*, 223:1–56, 1999. URL: `https://jeffe.cs.illinois.edu/pubs/pdf/survey.pdf`.

4 Pankaj K. Agarwal and Micha Sharir. Davenport-Schinzel Sequences and Their Geometric Applications. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 1–47. North Holland / Elsevier, 2000. `doi:10.1016/B978-044482537-7/50002-4`.

5 Boris Aronov, Mark de Berg, Jean Cardinal, Esther Ezra, John Iacono, and Micha Sharir. Subquadratic Algorithms for Some 3SUM-Hard Geometric Problems in the Algebraic Decision-Tree Model. *Comput. Geom.*, 109:101945, 2023. `doi:10.1016/J.COMGEO.2022.101945`.

6 Francis Avnaim and Jean-Daniel Boissonnat. Polygon Placement Under Translation and Rotation. *RAIRO Theor. Informatics Appl.*, 23(1):5–28, 1989. `doi:10.1051/ITA/1989230100051`.

7 Henry Spalding Baird. *Model-Based Image Matching Using Location.* MIT Press, 1984.

8 Luis Barba, Jean Cardinal, John Iacono, Stefan Langerman, Aurélien Ooms, and Noam Solomon. Subquadratic Algorithms for Algebraic 3SUM. *Discret. Comput. Geom.*, 61(4):698–734, 2019. Preliminary version in SoCG 2017. `doi:10.1007/S00454-018-0040-Y`.

9 Gill Barequet and Sariel Har-Peled. Polygon Containment and Translational Min-Hausdorff-Distance Between Segment Sets are 3SUM-Hard. *Int. J. Comput. Geom. Appl.*, 11(4):465–474, 2001. `doi:10.1142/S0218195901000596`.

10 Timothy M. Chan. Geometric Applications of a Randomized Optimization Technique. *Discret. Comput. Geom.*, 22(4):547–567, 1999. Preliminary version in SoCG 1998. `doi:10.1007/PL00009478`.

11 Timothy M. Chan. More Logarithmic-Factor Speedups for 3SUM, (median, +)-Convolution, and Some Geometric 3SUM-Hard Problems. *ACM Trans. Algorithms*, 16(1):7:1–7:23, 2020. `doi:10.1145/3363541`.

12 Bernard Chazelle. The Polygon Containment Problem. *Advances in Computing Research*, 1(1):1–33, 1983. URL: `https://www.cs.princeton.edu/čhazelle/pubs/PolygContainmentProb.pdf`.

13 L. Paul Chew and Klara Kedem. A Convex Polygon Among Polygonal Obstacles: Placement and High-Clearance Motion. *Comput. Geom.*, 3:59–89, 1993. `doi:10.1016/0925-7721(93)90001-M`.

14 Karen L. Daniels and Victor Milenkovic. Multiple Translational Containment. Part I: An Approximate Algorithm. *Algorithmica*, 19(1/2):148–182, 1997. doi:10.1007/PL00014415.

15 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2008. URL: https://www.worldcat.org/oclc/227584184.

16 Matthew Dickerson and Daniel Scharstein. Optimal Placement of Convex Polygons to Maximize Point Containment. In *Proc. 7th ACM-SIAM Symposium on Discrete Algorithm (SODA)*, pages 114–121, 1996. URL: http://dl.acm.org/citation.cfm?id=313852.313899.

17 Taekang Eom, Seungjun Lee, and Hee-Kap Ahn. Largest Similar Copies of Convex Polygons Amidst Polygonal Obstacles. *CoRR*, abs/2012.06978, 2020. arXiv:2012.06978.

18 Allan Grønlund and Seth Pettie. Threesomes, Degenerates, and Love Triangles. *J. ACM*, 65(4):22:1–22:25, 2018. doi:10.1145/3185378.

19 Daniel M. Kane, Shachar Lovett, and Shay Moran. Near-Optimal Linear Decision Trees for k-SUM and Related Problems. *J. ACM*, 66(3):16:1–16:18, 2019. doi:10.1145/3285953.

20 Marvin Künnemann and André Nusser. Polygon Placement Revisited: (Degree of Freedom + 1)-SUM Hardness and an Improvement via Offline Dynamic Rectangle Union. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3181–3201, 2022. doi:10.1137/1.9781611977073.124.

21 D. T. Lee. Interval, Segment, Range, and Priority Search Trees. In Dinesh P. Mehta and Sartaj Sahni, editors, *Handbook of Data Structures and Applications*. Chapman and Hall/CRC, 2004. doi:10.1201/9781420035179.CH18.

22 Seungjun Lee, Taekang Eom, and Hee-Kap Ahn. Largest Triangles in a Polygon. *Comput. Geom.*, 98:101792, 2021. doi:10.1016/J.COMGEO.2021.101792.

23 Nimrod Megiddo. Applying Parallel Computation Algorithms in the Design of Serial Algorithms. *J. ACM*, 30(4):852–865, 1983. doi:10.1145/2157.322410.

24 Victor Milenkovic. Translational Polygon Containment and Minimal Enclosure Using Linear Programming Based Restriction. In *Proc. 28th ACM Symposium on Theory of Computing (STOC)*, pages 109–118, 1996. doi:10.1145/237814.237840.

25 Victor Milenkovic. Multiple Translational Containment. Part II: Exact Algorithms. *Algorithmica*, 19(1/2):183–218, 1997. doi:10.1007/PL00014416.

26 Victor Milenkovic. Rotational Polygon Containment and Minimum Enclosure Using Only Robust 2D Constructions. *Comput. Geom.*, 13(1):3–19, 1999. doi:10.1016/S0925-7721(99)00006-1.

27 Joseph O'Rourke, Subhash Suri, and Csaba D Tóth. Polygons. In *Handbook of Discrete and Computational Geometry*, pages 787–810. CRC Press, 2017. URL: http://www.csun.edu/~ctoth/Handbook/chap30.pdf.

28 Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction*. Texts and Monographs in Computer Science. Springer, 1985. doi:10.1007/978-1-4612-1098-6.

29 Micha Sharir. A Near-Linear Algorithm for the Planar 2-Center Problem. *Discret. Comput. Geom.*, 18(2):125–134, 1997. Preliminary version in SoCG 1996. doi:10.1007/PL00009311.

30 Micha Sharir and Sivan Toledo. External Polygon Containment Problems. *Computational Geometry*, 4(2):99–118, 1994. Preliminary version in SoCG 1991. doi:10.1016/0925-7721(94)90011-6.