Kinetic Reverse k-Nearest Neighbor Problem [*]

Zahed Rahmati[†] Valerie King[‡] Sue Whitesides[§]

Abstract

This paper provides the first solution to the kinetic reverse k-nearest neighbor (RkNN) problem in $\mathbb{R}^d$, which is defined as follows: Given a set P of n moving points in arbitrary but fixed dimension d, an integer k, and a query point $q \notin P$ at any time t, report all the points $p \in P$ for which q is one of the k-nearest neighbors of p.

1 Introduction

The *reverse k-nearest neighbor* (RkNN) problem is a popular variant of the k-nearest neighbor (kNN) problem and asks for the influence of a query point on a point set. Unlike the kNN problem, the exact number of reverse k-nearest neighbors of a query point is not known in advance. The RkNN problem is formally defined as follows: Given a set P of n points in $\mathbb{R}^d$, an integer k, $1 \leq k \leq n-1$, and a query point $q \notin P$, find the set RkNN(q) of all p in P for which q is one of k-nearest neighbors of p. Thus RkNN$(q) = \{p \in P : \ |pq| \leq |pp_k|\}$, where $|.|$ denotes Euclidean distance, and p_k is the k^{th} nearest neighbor of p among the points in P. The *kinetic RkNN* problem is to answer RkNN queries on a set P of moving points, where the trajectory of each point $p \in P$ is a function of time. Here, we assume the trajectories are polynomial functions of maximum degree bounded by some constant s.

Related work. The reverse k-nearest neighbor problem was first posed by Korn and Muthukrishnan [14] in the database community, and then considered extensively in this community due to its many applications, *e.g.*, decision support systems, profile-based marketing, traffic networks, business location planning, clustering and outlier detection, and molecular biology [14, 15, 16]. The reverse k-nearest neighbor queries for a set of continuously moving objects has also attracted the attention of the database community; see [9] and references therein. Examples of moving objects

[*] This work was partially supported by a British Columbia Graduate Student Fellowship and by NSERC discovery grants.

[†] Department of Computer Science, University of Victoria, Canada, rahmati@uvic.ca

[‡] Department of Computer Science, University of Victoria, Canada, val@uvic.ca

[§] Department of Computer Science, University of Victoria, Canada sue@uvic.ca

include players in multi-player game environments, soldiers in a battlefield, tourists in dangerous environments, and mobile devices in wireless ad-hoc networks.

To our knowledge, in computational geometry, there exist two data structures [17, 10] that give solutions to the RkNN problem. Both of these solutions answer RkNN queries for a set P of stationary points and both only work for $k = 1$. Maheshwari *et al.* (2002) [17] gave a data structure to solve the R1NN problem in $\mathbb{R}^2$. Their data structure, which supports insertions and deletions of points, creates an arrangement of largest empty circles centered at the points of P and answers R1NN queries by point location in the arrangement. Their data structure uses $O(n)$ space and $O(n \log n)$ preprocessing time, and an R1NN query can be answered in time $O(\log n)$. Cheong *et al.* (2011) [10] considered the R1NN problem in fixed dimension $\mathbb{R}^d$, where $d = O(1)$. Their method, which uses a compressed quadtree, partitions space into cells such that each cell contains a small number of candidate points. To answer an R1NN query, their solution finds a cell that contains the query point and then checks all the points in the cell. Their approach uses $O(n)$ space and $O(n \log n)$ preprocessing time, and can answer an R1NN query in $O(\log n)$ time; it seems that the approach by Cheong *et al.* can be extended to answer RkNN queries with preprocessing time $O(kn \log n)$, space $O(kn)$, and query time $O(\log n + k)$.

For a set P of n stationary points, one can report all the 1-nearest neighbors in time $O(n \log n)$ [20], and all the k-nearest neighbors, for any $k \geq 1$, in time $O(kn \log n)$ [13], where the neighbors are reported in order of increasing distance from each point; reporting the unordered set takes time $O(n \log n + kn)$ [6, 11, 13].

For a set of moving points, there are two kinetic data structures [2, 19] to maintain all the k-nearest neighbors, but they only work for $k = 1$.

Our contribution. We provide the *first* solution to the kinetic RkNN problem for *any* $k \geq 1$ in *any* fixed dimension d. To answer an RkNN query for a query point $q \notin P$ at any time t, we partition the d-dimensional space into a constant number of cones around q, and then among the points of P in each cone, we examine the k points having shortest projections on the cone axis. We obtain $O(k)$ candidate points for q such that q might be one of their k-nearest neighbors at time t. To check which if any of these candidate points is a reverse k-nearest neighbor of q, we maintain the k^{th} nearest neighbor p_k of each point $p \in P$ over time. By checking whether $|pq| \leq |pp_k|$ we can easily check whether a candidate point p is one of the reverse k-nearest neighbors of q at time t.

For a set P of n continuously moving points in $\mathbb{R}^d$, where the trajectory of each point is a polynomial function of at most constant degree s, we provide a simple kinetic approach to answer RkNN queries on the moving points. In the preprocessing step, we introduce a method for reporting all the k-nearest neighbors for all the points $p \in P$ in order of increasing distance from p. For $k = \Omega(\log^{d-1} n)$, both our method and the method of Dickerson and Eppstein [13] give the same complexity, but in our view, our method is simpler in practice.

In order to answer RkNN queries, our kinetic approach maintains all the k-nearest neighbors over time. This is the first KDS for maintenance of all the k-nearest neighbors in $\mathbb{R}^d$, for any $k \geq 1$. Our KDS uses $O(n \log^d n + kn)$ space and $O(n \log^d n + kn \log n)$ preprocessing time, and processes $O(\phi(s,n) * n^2)$ events, each in amortized time $O(\log n)$. Here, $\phi(s,n)$ is the complexity of the k-level of a set of n partially-defined polynomial functions, such that each pair of them intersects at most s times. The current bounds on $\phi(s,n)$ are as follows.

$$\phi(s,n) = \begin{cases} O(n^{3/2} \log n), & \text{for } s = 2 \text{ [8];} \\ O(n^{5/3} \text{poly} \log n), & \text{for } s = 3 \text{ [7];} \\ O(n^{31/18} \text{poly} \log n), & \text{for } s = 4 \text{ [7];} \\ O(n^{161/90-\delta}), & \text{ for } s = 5, \text{ for some constant } \delta > 0 \text{ [8];} \\ O(n^{2-1/2s}), & \text{for odd } s \text{ [7];} \\ O(n^{2-1/2(s-1)}), & \text{for even } s \text{ [7].} \end{cases}$$

At any time t, an RkNN query can be answered in time $O(\log^d n + k \log\log n)$. Note that if an event occurs at the same time t, we first spend amortized time $O(\log n)$ to update all the k-nearest neighbors, and then we answer the query.

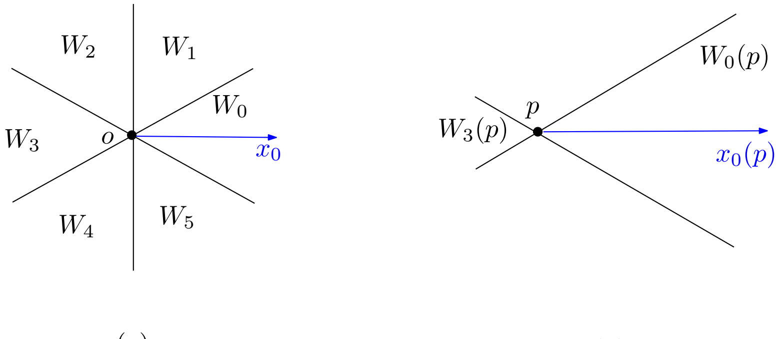
Outline. Section 2 provides two key lemmas, and in fact introduces a new super-graph, namely the *k-Semi-Yao graph*, of the k-nearest neighbor graph. In Section 3, we show how to report all the k-nearest neighbors. Section 4 gives a (kinetic) data structure for answering RkNN queries on moving points, where the trajectory of each point is a bounded-degree polynomial. Also included in this section is an analysis of our kinetic data structure in terms of the kinetic data structure performance criteria. Section 5 concludes.

2 Key Lemmas

Partition the plane around the origin o into six wedges, $W_0, ..., W_5$, each of angle $\pi/3$ (see Figure 1(a)). Denote by $W_l(p)$ the translation of wedge W_l, $0 \leq l \leq 5$, such that its apex moves from o to point p (see Figure 1(b)). Denote by x_l (resp. $x_l(p)$) the vector along the bisector of W_l (resp. $W_l(p)$) directed outward from the apex at o (resp. p). Denote the reflection of $W_l(p)$ through p by $W_{l'}(p)$. Note that $l' = (l+3) \bmod 6$; see Figure 1(b).

Consider the i^{th} nearest neighbor p_i of p. Denote by $L(P \cap W_l(p_i))$ the list of the points in $P \cap W_l(p_i)$, sorted by increasing order of their x_l-coordinates (projections). The following lemma provides a key insight.

Lemma 1 *Let p_i be the i^{th} nearest neighbor of p among a set P of points in $\mathbb{R}^2$, and let $W_l(p_i)$ be the wedge of p_i that contains p. Then point p is among the first i points in $L(P \cap W_l(p_i))$.*



(a) (b)

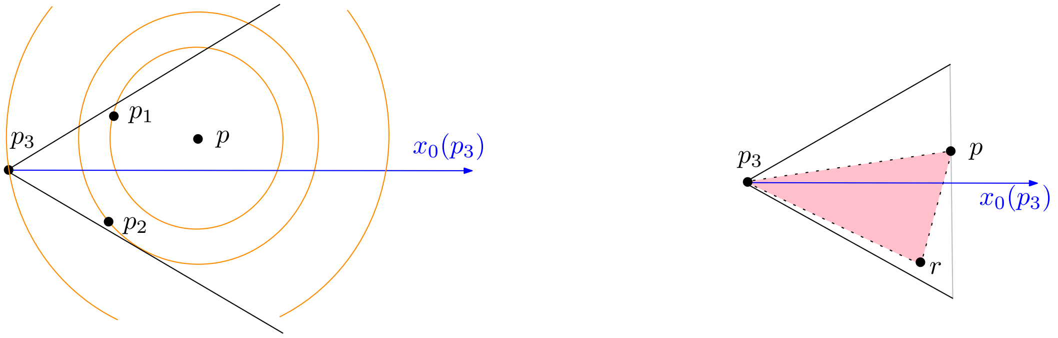
Figure 1: (a) A Partition of the plane into six wedges with common apex at o. (b) A translation of W_0 that moves apex to p. The wedge $W_0(p)$ is the reflection through p of $W_3(p)$ and vise-versa.

Proof. Let $P' = P \backslash \{p_1, ..., p_{i-1}\}$. Then the point p_i is the closest point to p among the points in P'; see Figure 2(a) below. We now prove by contradiction that the point p has the minimum x_l-coordinate among the points in $P' \cap W_l(p_i)$: Assume there is a point $r \in P$ inside the wedge $W_l(p_i)$ whose x_l-coordinate is less than the x_l-coordinate of p; see Figure 2(b) for an example where $i = 3$. Consider the triangle pp_ir. Since p_i is the closest point to p among the points in P', $|pp_i| < |pr|$ which implies that the angle $\angle pp_ir > \angle prp_i$. This is a contradiction, because $\angle pp_ir \leq \pi/3$ and $\angle prp_i > \pi/3$.

Now we add the points $p_1, ..., p_{i-2}$, and p_{i-1} to the point set P'. Consider the worst case scenario that all these $i - 1$ points insert inside the wedge $W_l(p_i)$, and that the x_l-coordinates of all these points are less than the x_l-coordinate of p. Then the point p is still among the first i points in the sorted list $L(P \cap W_l(p_i))$. □

The *k-nearest neighbor graph* (k-NNG) of a point set P is constructed by connecting each point in P to all its k-nearest neighbors. If we connect each point $p \in P$ to the first k points in the sorted list $L(P \cap W_l(p))$, for $l = 0, ..., 5$, we obtain what we call the *k-Semi-Yao graph* (k-SYG). Lemma 1 gives a necessary condition for p_i to be the i^{th} nearest neighbor of p: the point p is among the first i points in $L(P \cap W_l(p_i))$, where l is such that $p \in W_l(p_i)$. Therefore, the edge set of the k-SYG covers the edges of the k-NNG. In summary, we have the following.

Lemma 2 *The k-NNG of a set P of points in* $\mathbb{R}^2$ *is a subgraph of the k-SYG of the set P.*



(a)

(b)

Figure 2: Point p_3 is the 3rd nearest neighbor of p. After deleting the points p_1 and p_2, point p_3 is the closest point to p; among the points in $W_0(p_3)$, p has the minimum length projection on the bisector $x_0(p_3)$.

3 Reporting All k-Nearest Neighbors

Here we give a simple method for reporting all the k-nearest neighbors via a construction of the k-SYG.

Let C be a *right circular cone* in $\mathbb{R}^d$ with opening angle θ with respect to some given unit vector v. Thus C is the set of points $x \in \mathbb{R}^d$ such that the angle between $\overrightarrow{ox}$ and $\overrightarrow{v}$ is at most $\theta/2$. The angle between any two rays inside C emanating from the apex o is at most θ. From now on, we assume $\theta = \pi/3$.

Now consider a *polyhedral cone* inscribed in the right circular cone C where the polyhedral cone is formed by the intersection of d distinct half-spaces, bounded by $f_1, ..., f_d$, passing through the apex of C. Assuming d is arbitrary but fixed, the d-dimensional space around the origin o can be tiled by a constant number of polyhedral cones $W_0, ..., W_{c-1}$ [1, 2]. Denote by C_l the associated right circular cone of the polyhedral cone W_l. Let x_l be the vector in the direction of the symmetry of C_l. Denote by $W_l(p)$ the translation of the wedge (polyhedral cone) W_l where o moves to p.

A similar approach and analysis as that in Section 2 can be easily used to state (key) Lemmas 1 and 2 for a set of points in $\mathbb{R}^d$.

To construct the k-SYG efficiently, we need a data structure to perform the following operation efficiently: For each $p \in P$ and any of its wedges $W_l(p)$, $0 \leq l \leq c-1$, find the first k points in $L(P \cap W_l(p))$. Such an operation can be performed by using *range tree* data structures. For each wedge W_l with apex at origin o, we construct an associated d-dimensional range tree $\mathcal{T}_l$ as follows.

Consider a particular wedge W_l with apex at o. The wedge W_l is the intersection of d half-spaces $f_1^+, ..., f_d^+$ bounded by $f_1, ..., f_d$ (see Figure 3). Let $\hat{u}_j$ denote the

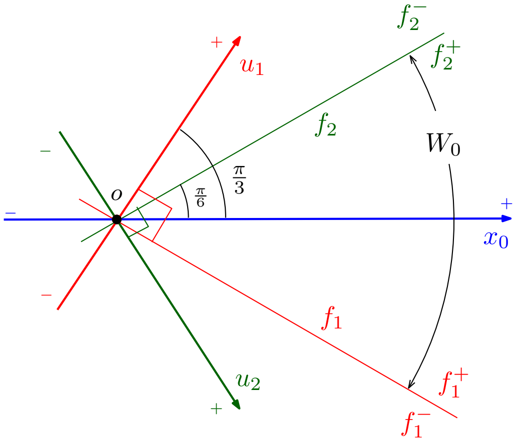


Figure 3: The wedge W_0 in $\mathbb{R}^2$ is bounded by f_1 and f_2. The coordinate axes u_1 and u_2 are orthogonal to f_1 and f_2.

normal to f_j pointing to f_j^+. We define d coordinate axes u_j, $j = 1, ..., d$, through $\hat{u_j}$, where $\hat{u_j}$ gives the respective directions of increasing u_j-coordinate values.

The range tree $\mathcal{T}_l$ is a regular d-dimensional range tree based on the u_j-coordinates, $j = 1, ..., d$. The points at level j are sorted at the leaves according to their u_j-coordinates (for more details about range trees, see Chapter 5 of [5]). From Theorem 5.8 in [5], any d-dimensional range tree, *e.g.*, $\mathcal{T}_l$, uses $O(n \log^{d-1} n)$ space and can be constructed in time $O(n \log^{d-1} n)$; for any point $r \in \mathbb{R}^d$, the points of P inside the query wedge $W_l(r)$ whose sides are parallel to f_j, $j = 1, ..., d$, can be reported in time $O(\log^d n + z)$, where z is the cardinality of the set $P \cap W_l(r)$. In particular, in time $O(\log^d n)$ one can determine a set of $O(\log^d n)$ internal nodes v at level d of $\mathcal{T}_l$, such that $P \cap W_l(r) = \bigcup_v P(v)$, where $P(v)$ is the set of points at the leaves of subtree rooted at v.

Now we add a new level to $\mathcal{T}_l$, based on the coordinate x_l. Let $\mathcal{C}_l(p)$ be the set of the first k points in $L(P \cap W_l(p))$. To find $\mathcal{C}_l(p)$ in an efficient time, we use the level $d+1$ of $\mathcal{T}_l$, which is constructed as follows: For each internal node v at level d of $\mathcal{T}_l$, we create a list $L(P(v))$ sorted by increasing order of x_l-coordinates of the points in $P(v)$. For the set P of n points in $\mathbb{R}^d$, the range tree $\mathcal{T}_l$, which now is a $(d+1)$-dimensional range tree, uses $O(n \log^d n)$ space and can be constructed in time $O(n \log^d n)$.

The following lemma establishes the processing time for obtaining a $\mathcal{C}_l(p)$.

Lemma 3 *Given $\mathcal{T}_l$, the set $\mathcal{C}_l(p)$ can be found in time $O(\log^d n + k \log\log n)$.*

Proof. The proof is by construction. Recall that the set $P \cap W_l(p)$ is the union of $O(\log^d n)$ sets $P(v)$, where v ranges over internal nodes at level d of $\mathcal{T}_l$. Consider the associated sorted lists $L(P(v))$.

We construct a priority queue on the first elements of these $O(\log^d n)$ sorted lists $L(P(v))$ in time $O(\log^d n)$.

By repeating the following two steps k times we can find $\mathcal{C}_l(p)$:

- Delete the element $\hat{p}$ with highest priority from the priority queue, and
- insert the next element into the priority queue from the sorted list $L(P(v_j))$, where v_j is such that $\hat{p} \in P(v_j)$.

Since d is fixed and the size of the priority queue is $O(\log^d n)$, all together these k iterations take $O(k \log \log n)$ time. □

By Lemma 3, we can find all the $\mathcal{C}_l(p)$, for all the points $p \in P$. This gives the following lemma.

Lemma 4 *Using a data structure of size $O(n \log^d n)$, the edges of the k-SYG of a set of n points in fixed dimension d can be reported in time $O(n \log^d n + kn \log \log n)$.*

Next, suppose we are given the k-SYG and we want to report all the k-nearest neighbors. Let E_p be the set of edges incident to the point p in the k-SYG. By sorting these edges in non-decreasing order according to their Euclidean lengths, which can be done in time $O(|E_p| \log |E_p|)$, we can find the k-nearest neighbors of p ordered by increasing distance from p. Since the number of edges in the k-SYG is $O(kn)$ and each edge pp' belongs to exactly two sets E_p and $E_{p'}$, the time to find all the k-nearest neighbors, for all the points $p \in P$, is $\sum_p O(|E_p| \log |E_p|) = O(kn \log n)$.

From the above discussion and Lemmas 2 and 4, the following results.

Theorem 1 *For a set of n points in fixed dimension d, our data structure can report all the k-nearest neighbors, in order of increasing distance from each point, in time $O(n \log^d n + kn \log n)$. The data structure uses $O(n \log^d n + kn)$ space.*

4 RkNN Queries on Moving Points

We are given a set P of n continuously moving points, where the trajectory of each point in P is a polynomial function of bounded degree s. To answer RkNN queries on the moving points, we must keep a valid range tree and track all the k-nearest neighbors during the motion. This section first shows how to maintain a (ranked-based) range tree, and then provides a KDS for maintenance of the k-SYG, which in fact gives a supergraph of the k-NNG over time. Using the kinetic k-SYG, we can easily maintain all the k-nearest neighbors over time. Finally we show how to answer RkNN queries on the moving points.

Kinetic RBRT. Let u_j, $1 \leq j \leq d$, be the coordinate axis orthogonal to the half-space f_j of the wedge W_l, $0 \leq l \leq c-1$ (see Figure 3). Abam and de Berg [1] introduced a variant of the range tree, namely the *ranked-based range tree* (RBRT), which has the following properties. Denote by $\mathcal{T}_l$ the RBRT corresponding to the wedge W_l.

- $\mathcal{T}_l$ can be described as a set of pairs $\Psi_l = \{(B_1, R_1), ..., (B_m, R_m)\}$ such that:
  - For any two points p and q in P where $q \in W_l(p)$, there is a unique pair $(B_i, R_i) \in \Psi_l$ such that $p \in B_i$ and $q \in R_i$.
  - For any pair $(B_i, R_i) \in \Psi_l$, if $p \in B_i$ and $q \in R_i$, then $q \in W_l(p)$ and $p \in W_{l'}(q)$; here $W_{l'}(q)$ is the reflection of $W_l(q)$ through q.

  The Ψ_l is called a *cone separated pair decomposition* (CSPD) for P with respect to W_l. Each pair (B_i, R_i) is generated from an internal node v at level d of the RBRT $\mathcal{T}_l$.

- Each point $p \in P$ is in $O(\log^d n)$ pairs of (B_i, R_i), which means that the number of elements of all the pairs (R_i, B_i) is $O(n \log^d n)$.

- For any point $p \in P$, all the sets B_i (resp. R_i) where $p \in B_i$ (resp. $p \in R_i$) can be found in time $O(\log^d n)$.

- The set $P \cap W_l(p)$ is the union of $O(\log^d n)$ sets R_i, where $p \in B_i$.

- When the points are moving, $\mathcal{T}_l$ remains unchanged as long as the order of the points along axes u_j, $1 \leq j \leq d$, remains unchanged.

- When a u-swap event occurs, meaning that two points exchange their u_j-order, the RBRT $\mathcal{T}_l$ can be updated in worst-case time $O(\log^d n)$ without rebalancing operations.

4.1 Kinetic k-SYG

Here we give a KDS for the k-SYG, for any $k \geq 1$, extending [18].

To maintain the k-SYG, we must track the set $\mathcal{C}_l(p)$ for each point $p \in P$. So, for each $1 \leq i \leq m$, we need to maintain a sorted list $L(R_i)$ of the points in R_i in ascending order according to their x_l-coordinates over time. Note that each set R_i is some $P(v)$, the set of points at the leaves of the subtree rooted at some internal node v at level d of $\mathcal{T}_l$. To maintain these sorted lists $L(R_i)$, we add a new level to the RBRT $\mathcal{T}_l$; the points at the new level are sorted at the leaves in ascending order according to their x_l-coordinates. Therefore, in the modified RBRT $\mathcal{T}_l$, in addition to the u-swap events, we handle new events, called *x-swap events*, when two points exchange their x_l-order. The modified RBRT $\mathcal{T}_l$ behaves like a $(d+1)$-dimensional RBRT. From the last property of an RBRT above, when a u-swap event or an x-swap event occurs, the RBRT $\mathcal{T}_l$ can be updated in worst-case time $O(\log^{d+1} n)$.

Denote by $\ddot{p}_{l,k}$ the k^{th} point in $L(P \cap W_l(p))$. To track the sets $\mathcal{C}_l(p)$, for all the points $p \in P$, we need to maintain the following over time.

- A set of $d+1$ *kinetic sorted lists* $L_j(P)$, $j = 1, ..., d$, and the $L_l(P)$ of the point set P. We use these kinetic sorted lists to track the order of the points in the coordinates u_j and x_l, respectively.
- For each B_i, a sorted list $L(B'_i)$ of the points in B'_i, where $B'_i = \{(p, \ddot{p}_{l,k}) | \; p \in B_i\}$. The order of the points in $L(B'_i)$ is according to a *label* of the second points $\ddot{p}_{l,k}$. This sorted list $L(B'_i)$ is used to answer the following query efficiently: Given a query point q and a B_i, find all the points $p \in B_i$ such that $\ddot{p}_{l,k} = q$.
- The k^{th} point $r_{i,k}$ in the sorted list $L(R_i)$. We track the values $r_{i,k}$ in order to make necessary changes to the k-SYG when an x-swap event occurs.

Handling u-swap events. W.l.o.g., let $q \in W_l(p)$ before the event. When a u-swap event between p and q occurs, the point q moves outside the wedge $W_l(p)$; after the event, $q \notin W_l(p)$. Note that the changes that occur in the k-SYG are the deletions and insertions of the edges incident to p inside the wedge $W_l(p)$.

Whenever two points p and q exchange their u_j-order, we do the following updates.

- We update the kinetic sorted list $L_j(P)$. Each swap event in a kinetic sorted list can be handled in time $O(\log n)$.
- We update the RBRT $\mathcal{T}_l$ and if a point is deleted or inserted into a B_i, we update the sorted list $L(B'_i)$. Since each insertion/deletion to $L(B'_i)$ takes $O(\log n)$ time, and since each point is in $O(\log^d n)$ sets B_i, this takes $O(\log^{d+1} n)$ time.
- We update the values of $r_{i,k}$. After updating the RBRT $\mathcal{T}_l$, point q might be inserted or deleted from some R_i and change the values of $r_{i,k}$. So, for all R_i where $q \in R_i$, before and after the event, we do the following. We check whether the x_l-coordinate of q is less than or equal to the x_l-coordinate of $r_{i,k}$; if so, we take the successor or predecessor point of $r_{i,k}$ in $L(R_i)$ as the new value for $r_{i,k}$. This takes $O(\log^{d+1} n)$ time.
- We query to find $\mathcal{C}(p)$. By Lemma 3, this takes $O(\log^d n + k \log\log n)$ time.
- If we get a new value for $\ddot{p}_{l,k}$, we update all the sorted lists $L(B'_i)$ such that $p \in B_i$. This takes $O(\log^{d+1} n)$ time.

Considering the complexity of each step above, and assuming the trajectory of each point is a bounded degree polynomial, the following results.

Lemma 5 *Our KDS for maintenance of the k-SYG handles $O(n^2)$ u-swap events, each in worst-case time $O(\log^{d+1} n + k \log\log n)$.*

Handling x-swap events. When an x-swap event between two consecutive points p and q with p preceding q occurs, it does not change the elements of the pairs (B_i, R_i) of the CSPD Ψ_l. Such an event changes the k-SYG if both p and q are in the same $W_l(w)$, for some $w \in P$, and $w_{l,k} = p$.

We apply the following updates to our KDS when two points p and q exchange their x_l-order.

1. We update the kinetic sorted list $L_l(P)$; this takes $O(\log n)$ time.
2. We update the RBRT $\mathcal{T}_l$, which takes $O(\log^{d+1} n)$ time.
3. We find all the sets R_i where both p and q belong to R_i and such that $r_{i,k} = p$. Also, we find all the sets R_i where $r_{i,k} = q$. This takes $O(\log^d n)$ time.
4. For each R_i, we extract all the pairs $(w, \ddot{w}_{l,k})$ from the sorted lists $L(B'_i)$ such that $\ddot{w}_{l,k} = p$. Note that each change to the pair $(w, \ddot{w}_{l,k})$ is a change to the k-SYG.
5. For each w, we update all the sorted lists $L(B'_i)$ where $(w, \ddot{w}_{l,k}) \in B'_i$: we replace the previous value of $\ddot{w}_{l,k}$, which is p, by the new value q.

Denote by χ_k the number of exact changes to the k-SYG of a set of moving points over time. For each found R_i, the fourth step takes $O(\log n + \xi_i)$ time, where ξ_i is the number of pairs $(w, \ddot{w}_{l,k})$ such that $\ddot{w}_{l,k} = p$. For all these $O(\log^d n)$ sets R_i, this step takes $O(\log^{d+1} n + \sum_i \xi_i)$ time, where $\sum_i \xi_i$ is the number of exact changes to the k-SYG when an x-swap event occurs. Therefore, for all the $O(n^2)$ x-swap events, the total processing time for this step is $O(n^2 \log^{d+1} n + \chi_k)$.

The processing time for the fifth step is a function of χ_k. For each change to the k-SYG, this step spends $O(\log^{d+1} n)$ time to update the sorted lists $L(B'_i)$. Therefore, the total processing time for all the x-swap events in this step is $O(\chi_k * \log^{d+1} n)$.

From the above discussion and an upper bound for χ_k in Lemma 6, Lemma 7 results.

Lemma 6 *The number of changes to the k-SYG of a set of n moving points, where the trajectory of each point is a polynomial function of at most constant degree s, is $\chi_k = O(\phi(s,n) * n)$.*

Proof. Fix a point $p \in P$ and one of its wedges $W_l(p)$. There are $O(n)$ insertions/deletions into the wedge $W_l(p)$ over time. The x_l-coordinates of these points create $O(n)$ partial functions.

The k-SYG changes if a change to $\ddot{p}_{l,k}$ occurs. The number of all changes to $\ddot{p}_{l,k}$ is equal to $\phi(s,n)$, the complexity of the k-level of partially-defined polynomial functions of bounded degree s.

Therefore, considering all the $n = |P|$ points, the number of changes to the k-SYG is within a linear factor of $\phi(s,n)$: $\chi_k = O(\phi(s,n) * n)$. □

Lemma 7 *Our KDS for maintenance of the k-SYG handles $O(n^2)$ x-swap events with a total cost of $O(\phi(s,n) * n \log^{d+1} n)$.*

From Lemmas 5 and 7, the following theorem results.

Theorem 2 *For a set of n moving points in $\mathbb{R}^d$, where the trajectory of each point is a polynomial function of at most constant degree s, our k-SYG KDS uses $O(n \log^d n)$ space and handles $O(n^2)$ events with a total cost of $O(kn^2 \log\log n + \phi(s,n) * n \log^{d+1} n)$.*

4.2 Kinetic All k-Nearest Neighbors

Given a KDS for maintenance of the k-SYG (from Theorem 2), a supergraph of the k-NNG, this section shows how to maintain all the k-nearest neighbors over time. For maintenance of the k-nearest neighbors of each point $p \in P$, we only need to track the order of the edges incident to p in the k-SYG according to their Euclidean lengths. This can easily be done by using a kinetic sorted list. The following theorem summarizes the complexity of our kinetic approach.

Theorem 3 *For a set of n moving points in $\mathbb{R}^d$, where the trajectory of each point is a polynomial of at most constant degree s, our KDS for maintenance of all the k-nearest neighbors, ordered by distance from each point, uses $O(n \log^d n + kn)$ space and $O(n \log^d n + kn \log n)$ preprocessing time. Our KDS handles $O(\phi(s,n) * n^2)$ events, each in $O(\log n)$ amortized time.*

Proof. Let $E_p(t)$ be the set of edges incident to point $p \in P$ in the k-SYG at time t. Let $L(E_p(t))$ denote a kinetic sorted list that maintains the edges in $E_p(t)$ sorted by their Euclidean lengths.

Let m_p be the number of insertions/deletions to the set $E_p(t)$ over time. Since the cardinality of $E_p(t)$ is $O(n)$, each insertion into a kinetic sorted list $L(E_p(t))$ can cause $O(n)$ swaps. Each change, *e.g.*, inserting/deleting an edge pq, to the k-SYG creates two insertions/deletions in the kinetic sorted lists $L(E_p(t))$ and $L(E_q(t))$; this implies that $\sum_p m_p = O(\chi_k)$. By Lemma 6, the kinetic sorted lists handle a total of $O(n \sum_p m_p) = O(\phi(s,n) * n^2)$ events. Each event in a kinetic sorted list is handled in time $O(\log n)$. Thus from this and Theorem 2, the total processing time for swap events is $O(kn^2 \log\log n + \phi(s,n) * n \log^{d+1} n + \phi(s,n) * n^2 \log n) = O(\phi(s,n) * n^2 \log n)$. □

KDS performance criteria. The KDS framework [4] measures the performance of a KDS by four standard criteria, which we now apply to our KDS for maintenance of all the k-nearest neighbors in $\mathbb{R}^d$.

- **Efficiency:** This is the ratio of the number of events that a KDS processes to the number of exact changes to the attribute of interest over time. The exact number of changes for maintenance of all the k-nearest neighbors can be computed as follows. Fix a point $p \in P$. The distances of the $n-1$ points of $P \backslash \{p\}$ to p as functions of time create $2s$-intersecting curves, meaning that each pair intersects at most $2s$ times. The number of changes to the i^{th} nearest neighbor p_i of p equals $\Phi(2s, n-1)$, the complexity of the i-level of the $n-1$ $2s$-intersecting curves. Thus the number of changes to the k-nearest neighbors $p_1, ..., p_k$ of p is $O(\Phi(2s, n) * k)$. The total for all points $p \in P$ is $O(\Phi(2s, n) * kn)$. Since the number of events in our KDS is $O(\phi(s, n) * n^2)$, the efficiency of our KDS is $O(\frac{n}{k})$.

- **Responsiveness:** This is the cost of updating the KDS when an event occurs. In our KDS each event can be handled in amortized time $O(\log n)$. Thus the responsiveness of our KDS is $O(\log n)$ on average.

- **Locality:** The number of updates to a KDS when a point changes its trajectory gives the locality of the KDS. In our KDS, for each two consecutive elements in each of the kinetic sorted lists $L_j(P)$, $L_l(P)$, and $L(E_p(t))$, we have a boolean function of time, called a *certificate*. Each certificate has a failure time, the time when the two consecutive elements exchange their order. If a point changes its trajectory, we update a constant number of these certificates in the kinetic sorted lists $L_j(P)$ and $L_l(P)$. Since the number of edges in the k-SYG is $O(kn)$, if a point changes its trajectory, the number of updates to the certificates in the kinetic sorted lists $L(E_p(t))$ is $O(k)$ on average. Therefore, the locality of our KDS is $O(k)$ on average.

- **Compactness:** This is the number of certificates in the KDS. Since the number of certificates of the kinetic sorted lists $L_j(P)$ and $L_l(P)$ is $O(n)$, and the number of certificates of the kinetic sorted lists $L(E_p(t))$ is $O(kn)$, the compactness of our KDS is $O(kn)$.

Therefore, we can obtain the following.

Lemma 8 *In terms of the KDS performance criteria, the "efficiency", "responsiveness", "locality", and "compactness" of our KDS are $O(nk)$, $O(\log n)$ on average, $O(k)$ on average, and $O(kn)$, respectively.*

4.3 RkNN Queries

Suppose we are given a query point $q \notin P$ at some time t. To find the reverse k-nearest neighbors of q, we seek the points in $P \cap W_l(q)$ and find $\mathcal{C}_l(q)$, the set of the first k points in $L(P \cap W_l(q))$. The set $\cup_l \mathcal{C}_l(q)$ contains $O(k)$ candidate points for q such that q might be one of their k-nearest neighbors. In time $O(\log^d n)$ we can find a set of R_i where $P \cap W_l(q) = \sum_i R_i$. From Lemma 3, and since we have sorted lists $L(R_i)$ at level $d+1$ of $\mathcal{T}_l$, the $O(k)$ candidate points for the query point q can be found in worst-case time $O(\log^d n + k \log\log n)$. Now we check whether these candidate points are the reverse k-nearest neighbors of the query point q at time t or not; this can be easily done by application of Theorem 3, which in fact maintain the k^{th} nearest neighbor p_k of each $p \in P$. Therefore, checking a candidate point can be done in $O(1)$ time by comparing distance $|pq|$ to distance $|pp_k|$. This implies that checking which elements of $\mathcal{C}_l(q)$, for $l = 0, ..., c-1$, are reverse k-nearest neighbors of the query point q takes time $O(k)$.

If a query arrives at a time t that is simultaneous with the time when one of the $O(\phi(s,n) * n^2)$ events occurs, our KDS first spends time $O(\log n)$ in an amortized sense to handle the event, and then spends time $O(\log^d n + k \log\log n)$ to answer the query. Thus we have the following.

Theorem 4 *Consider a set P of n moving points in $\mathbb{R}^d$, where the trajectory of each one is a bounded-degree polynomial. The number of reverse k-nearest neighbors for a query point $q \notin P$ is $O(k)$. Our (kinetic) data structure uses $O(n \log^d n + kn)$ space and $O(n \log^d n + kn \log n)$ preprocessing time. At any time t, an RkNN query can be answered in time $O(\log^d n + k \log\log n)$. If an event occurs at time t, the KDS spends $O(\log n)$ time in an amortized sense on updating itself.*

5 Discussion and Conclusion

In the kinetic setting, where the trajectories of the points are polynomials of bounded degree, to answer the RkNN queries over time we have provided a KDS for maintenance of all the k-nearest neighbors. Our KDS is the first KDS for maintenance of all the k-nearest neighbors in $\mathbb{R}^d$, for any $k \geq 1$. It processes $O(\phi(s,n) * n^2)$ events, each in time $O(\log n)$ in an amortized sense. An open problem is to design a KDS that processes less than $O(\phi(s,n) * n^2)$ events.

Arya *et al.* [3] have a kd-tree implementation to approximate the nearest neighbors of a query point that is in use by practitioners [12] who have found challenging to implement the theoretical algorithms [6, 11, 13, 20]. Since to report all the k-nearest neighbors ordered by distance from each point our method uses multidimensional range trees, which can be easily implemented, we believe our method may be useful in practice.

Acknowledgments. We thank Timothy Chan for his remarks on the best current bounds on the complexity of the k-level of partially-defined bounded-degree polynomials.

References

[1] Abam, M.A., de Berg, M.: Kinetic spanners in $\mathbb{R}^d$. Discrete & Computational Geometry 45(4), 723–736 (2011)

[2] Agarwal, P.K., Kaplan, H., Sharir, M.: Kinetic and dynamic data structures for closest pair and all nearest neighbors. ACM Transactions on Algorithms 5, 4:1–37 (2008)

[3] Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM 45(6), 891–923 (1998)

[4] Basch, J., Guibas, L.J., Hershberger, J.: Data structures for mobile data. Journal of Algorithms 31, 1–19 (1999)

[5] Berg, M.d., Cheong, O., Kreveld, M.v., Overmars, M.: Computational Geometry: Algorithms and Applications. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd edn. (2008)

[6] Callahan, P.: Optimal parallel all-nearest-neighbors using the well-separated pair decomposition. In: Proceedings of the 34th Annual Symposium on Foundations of Computer Science (FOCS '93). pp. 332–340. IEEE Computer Society, Washington, DC, USA (1993)

[7] Chan, T.M.: On levels in arrangements of curves, ii: A simple inequality and its consequences. Discrete & Computational Geometry 34(1), 11–24 (2005)

[8] Chan, T.M.: On levels in arrangements of curves, iii: further improvements. In: Proceedings of the 24th annual Symposium on Computational Geometry (SoCG '08). pp. 85–93. ACM, New York, NY, USA (2008)

[9] Cheema, M.A., Zhang, W., Lin, X., Zhang, Y., Li, X.: Continuous reverse k nearest neighbors queries in euclidean space and in spatial networks. The VLDB Journal 21(1), 69–95 (2012)

[10] Cheong, O., Vigneron, A., Yon, J.: Reverse nearest neighbor queries in fixed dimension. International Journal of Computational Geometry and Applications 21(02), 179–188 (2011)

[11] Clarkson, K.L.: Fast algorithms for the all nearest neighbors problem. In: Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS '83). pp. 226–232. IEEE Computer Society, Washington, DC, USA (1983)

[12] Connor, M., Kumar, P.: Fast construction of k-nearest neighbor graphs for point clouds. IEEE Transactions on Visualization and Computer Graphics 16(4), 599–608 (2010)

[13] Dickerson, M.T., Eppstein, D.: Algorithms for proximity problems in higher dimensions. International Journal of Computational Geometry and Applications 5(5), 277–291 (1996)

[14] Korn, F., Muthukrishnan, S.: Influence sets based on reverse nearest neighbor queries. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00). pp. 201–212. ACM, New York, NY, USA (2000)

[15] Kumar, Y., Janardan, R., Gupta, P.: Efficient algorithms for reverse proximity query problems. In: Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '08). pp. 39:1–39:10. ACM, New York, NY, USA (2008)

[16] Lin, J., Etter, D., DeBarr, D.: Exact and approximate reverse nearest neighbor search for multimedia data. In: Proceedings of the 2008 SIAM International Conference on Data Mining (SDM '08). pp. 656–667. SIAM (2008)

[17] Maheshwari, A., Vahrenhold, J., Zeh, N.: On reverse nearest neighbor queries. In: Proceedings of the 14th Canadian Conference on Computational Geometry (CCCG '02). pp. 128–132 (2002)

[18] Rahmati, Z., Abam, M.A., King, V., Whitesides, S.: Kinetic data structures for the Semi-Yao graph and all nearest neighbors in $\mathbb{R}^d$. In: Proceedings of the 26th Canadian Conference on Computational Geometry (CCCG '14) (2014)

[19] Rahmati, Z., King, V., Whitesides, S.: Kinetic data structures for all nearest neighbors and closest pair in the plane. In: Proceedings of the 29th Symposium on Computational Geometry (SoCG '13). pp. 137–144. ACM, New York, NY, USA (2013)

[20] Vaidya, P.M.: An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. Discrete & Computational Geometry 4(2), 101–115 (1989)

Approximating the Maximum Overlap of Polygons under Translation[*]

Sariel Har-Peled[†] Subhro Roy[‡]

June 24, 2014

Abstract

Let P and Q be two simple polygons in the plane of total complexity n, each of which can be decomposed into at most k convex parts. We present an $(1-\varepsilon)$-approximation algorithm, for finding the translation of Q, which maximizes its area of overlap with P. Our algorithm runs in $O(cn)$ time, where c is a constant that depends only on k and ε.

This suggest that for polygons that are "close" to being convex, the problem can be solved (approximately), in near linear time.

1. Introduction

Shape matching is an important problem in databases, robotics, visualization and many other fields. Given two shapes, we want to find how similar (or dissimilar) they are. Typical problems include matching point sets by the Hausdorff distance metric, or matching polygons by the Hausdorff or Fréchet distance between their boundaries. See the survey by Alt and Guibas [AG00].

The maximum area of overlap is one possible measure for shape matching that is not significantly effected by noise. Mount *et al.* [MSW96] studied the behavior of the area of overlap function, when one simple polygon is translated over another simple polygon. They showed that the function is continuous and piece-wise polynomial of degree at most two. If the polygons P and Q have complexity m and n, respectively, the area of overlap function can have complexity of $\Theta(m^2n^2)$. Known algorithms to find the maximum of the function work by constructing the entire overlap function. It is also known that the problem is 3SUM-Hard [BH01b], that is, it is believed no subquadratic time algorithm is possible for the problem.

Approximating maximum overlap of general polygons. Cheong *et al.* [CEH07] gave a $(1-\varepsilon)$-approximation algorithm for maximizing the area of overlap under translation of one simple polygon over the other using random sampling techniques. However, the error associated with the algorithm is additive, and the algorithm runs in near quadratic time. Specifically, the error is an ε fraction of the area of the smaller of the two polygons. Under rigid motions, the running time deteriorates to being

[*] Work on this paper was partially supported by NSF AF awards CCF-0915984 and CCF-1217462.

[†] Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@uiuc.edu; http://sarielhp.org.

[‡] Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sroy9@illinois.edu; http://web.engr.illinois.edu/~sroy9/.

near cubic. More recently, Cheng and Lam [CL13] improved the running times, and can also handle rigid motions, and present a near linear time approximation algorithm if one of the polygons is convex.

Maximum overlap in the convex case under translations. de Berg *et al.* [dBCD+98] showed that finding maximum overlap translation is relatively easier in case of convex polygons. Specifically, the overlap function in this case is unimodal (as a consequence of the Brunn-Minkowski Theorem). Using this property, they gave a near linear time exact algorithm for computing the translation that maximizes the area of overlap of two convex polygons. The complexity of the graph of the overlap function is only $O(m^2 + n^2 + \min(m^2 n, mn^2))$ in this case. Alt *et al.* [AFRW98] gave a constant-factor approximation for the minimum area of the symmetric difference of two convex polygons.

Approximating maximum overlap in the convex case. As for $(1-\varepsilon)$-approximation, assuming that the two polygons are provided in an appropriate form (i.e., the vertices are in an array in their order along the boundary of the polygon), then one can get a sub-linear time approximation algorithm. Specifically, Ahn *et al.* [ACP+07] show an $(1-\varepsilon)$-approximation algorithm, with running time $O((1/\varepsilon)\log(n/\varepsilon))$ for the case of translation, and $O((1/\varepsilon)\log n + (1/\varepsilon)^2 \log 1/\varepsilon))$ for the case of rigid motions. (For a result using similar ideas in higher dimensions see the work by Chazelle *et al.* [CLM05].)

Overlap of union of balls. de Berg *et al.* [dBCG+04] considered the case where X and Y are disjoint unions of m and n unit disks, with $m \leq n$. They computed a $(1-\varepsilon)$ approximation for the maximal area of overlap of X and Y under translations in time $O((nm/\varepsilon^2)\log(n/\varepsilon))$. Cheong *et al.* [CEH07] gave an additive error ε-approximation algorithm for this case, with near linear running time.

Other relevant results. Avis *et al.* [ABT+96] computes the overlap of a polytope and a translated hyperplane in linear time, if the polytope is represented by a lattice of its faces. Vigneron [Vig14] presented $(1-\varepsilon)$-approximation algorithms for maximum overlap of polyhedra (in constant dimension) that runs in polynomial time. Ahn *et al.* [ACKY14] approximates the maximum overlap of two convex polytopes in three dimensions under rigid motions. Ahn *et al.* [ACR13] approximates the maximum overlap of two polytopes in $\mathbb{R}^d$ under translation in $O\left(n^{\lfloor d/2 \rfloor + 1} \log^d n\right)$ time.

Our results

As the above indicates, there is a big gap between the algorithms known for the convex and non-convex case. Our work aims to bridge this gap, showing that for “close” to convex polygons, under translation, the problem can be solved approximately in near linear time.

Specifically, assume we are given two polygons P and Q of total complexity n, such that they can be decomposed into k convex parts, we show that one can $(1-\varepsilon)$-approximate the translation of Q, which maximizes its area of overlap with P, in linear time (for k and ε constants). The translation returned has overlap area which is at least $(1-\varepsilon)\mu_{\max}(\mathsf{P}, \mathsf{Q})$, where $\mu_{\max}(\mathsf{P}, \mathsf{Q})$ is the maximum area of overlap of the given polygons.

Approach. We break the two polygons into a minimum number of convex parts. We then approximate the overlap function for each pair of pieces (everywhere). This is required as one cannot just approximate the two polygons (as done by Ahn *et al.* [ACP+07]) since the optimal solution does not realize the maximum overlap of each pair of parts separately, and the alignment of each pair of parts might be arbitrary.

To this end, if the two convex parts are of completely different sizes, we approximate the smaller part, and approximate the overlap function by taking slices (i.e., level sets) of the overlap function. In the other case, where the two parts are "large", which is intuitively easier, we can approximate both convex parts, and then the overlap function has constant complexity. Finally, we overlap all these functions together, argue that the overlap has low complexity, and find the maximum area of overlap.

Our approach has some overlap in ideas with the work of Ahn *et al.* [ACP+07]. In particular, a similar distinction between large and small overlap, as done in Section 4.1 and Section 4.2 was already done in [ACP+07, Theorem 17].

Why the "naive" solution fails? The naive solution to our problem is to break the two polygons into k convex polygons, and then apply to each pair of them the approximation of Ahn *et al.* [ACP+07]. Now, just treat the input polygon as the union of their respective approximations, and solve problem using brute force approach. This fails miserably as the approximation of Ahn *et al.* [ACP+07] captures only the maximum overlap of the two polygons. It does not, and can not, approximates the overlap if two convex polygons are translated such that their overlap is "far" from the maximum configuration, especially if the two polygons are of different sizes. This issue is demonstrated in more detail in the beginning of Section 4.1. A more detailed counterexample is presented in Appendix B.

Paper organization. We start in Section 2 by defining formally the problem, and review some needed results. In Section 3, we build some necessary tools. Specifically, we start in Section 3.1 by observing that one can get $O(1/\varepsilon)$ approximation of a convex polygon, where the error is an ε-fraction of the width of the polygon. In Section 3.2, we show how to compute a level set of the overlap function of two convex polygons efficiently. In Section 3.3, we show that, surprisingly, the polygon formed by the maximum overlap of two convex polygons, contains (up to scaling by a small constant and translation) the intersection of any translation of these two convex polygons. Among other things this implies an easy linear time constant factor approximation for the maximum overlap (which also follows, of course, by the result of Ahn *et al.* [ACP+07]). In Section 4, we present the technical main contribution of this paper, showing how to approximate, by a compact representation that has roughly linear complexity, the area overlap function of two convex polygons. In Section 5 we put everything together and present our approximation algorithm for the non-convex case.

2. Preliminaries

For any vector $\mathsf{t} \in \mathbb{R}^2$ and a set Q, let $\mathsf{t} + \mathsf{Q}$ denote the translation of Q by t; formally, $\mathsf{t} + \mathsf{Q} = \left\{ \mathsf{t} + \mathsf{q} \,\middle|\, \mathsf{q} \in \mathsf{Q} \right\}$. Also let $\mu(\mathsf{P}, \mathsf{Q}) = \mathrm{area}(\mathsf{P} \cap \mathsf{Q})$, which is the ***area of overlap*** of sets P and Q. We are interested in the following problem.

Problem 2.1. *We are given two polygons* X *and* Y *in the plane, such that each can be decomposed into at most* k *convex polygons. The task is to compute the translation* t *of* Y*, which maximizes the area of overlap between* X *and* $\mathsf{t} + \mathsf{Y}$*. Specifically our purpose is to approximate the quantity*

$$\mu_{\max}(\mathsf{X}, \mathsf{Y}) = \max_{\mathsf{t} \in \mathbb{R}^2} \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y}).$$

For a polygon P, let $|\mathsf{P}|$ denote the number of vertices of P. For $\mathsf{X}, \mathsf{Y} \subseteq \mathbb{R}^d$, the set X is ***contained under translation*** in Y, denoted by $\mathsf{X} \sqsubseteq \mathsf{Y}$, if there exists $\vec{x}$ such that $\vec{x} + \mathsf{X} \subseteq \mathsf{Y}$.

Unimodal. A function $f : \mathbb{R} \to \mathbb{R}$ is ***unimodal***, if there is a value α, such that f is monotonically increasing (formally, non-decreasing) in the range $[-\infty, \alpha]$, and f is monotonically decreasing (formally, non-increasing) in the interval $[\alpha, +\infty]$.

From width to inner radius. For a convex polygon P, the ***width*** of P, denoted by $\omega(\mathsf{P})$, is the minimum distance between two parallel lines that enclose P.

Lemma 2.2 ([GK92]). *For a convex shape* X *in the plane, we have that the largest disk enclosed inside* X*, has radius at least* $\mathrm{width}(\mathsf{X})\,/2\sqrt{3}$.

Convex Decomposition of Simple Polygons. A vertex of a polygon is a ***notch*** if the internal angle at this vertex is reflex (i.e. $> 180°$). For a non-convex polygon P with n vertices and r notches, Keil and Snoeyink [KS02] solves the minimal convex decomposition problem in $O(n + r^2 \min(r^2, n))$ time, that is, they compute a decomposition of P into minimum number of convex polygons. Observe, that if the number of components in the minimum convex decomposition is k, the number of notches r is upper bounded by $2k$.

Scaling similarity between polygons. For two convex polygons X and Y, let us define their ***scaling similarity***, denoted by $\mathrm{ssim}(\mathsf{X}, \mathsf{Y})$, as the minimum number $\alpha \geq 0$, such that $\mathsf{X} \sqsubseteq \alpha\mathsf{Y}$. Using low-dimensional linear programming, one can compute $\mathrm{ssim}(\mathsf{X}, \mathsf{Y})$ in linear time. In particular, the work by Sharir and Toledo [ST94] implies the following.

Lemma 2.3 (ssim). *Given two convex polygons* X *and* Y *of total complexity* n*, one can compute, in linear time,* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y})$*, and the translation that realizes it.*

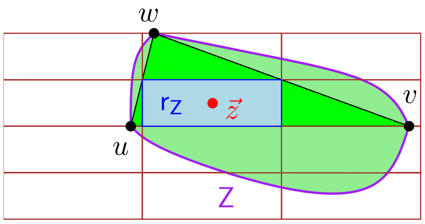
3. Building blocks

3.1. A better convex approximation in the plane

Let B be the minimum volume bounding box of some bounded convex set $K \subseteq \mathbb{R}^d$. We have that $v + c_d B \subseteq K \subseteq B$ [Har11], for some vector v and a constant c_d which depends only on the dimension d. This approximation can be computed in $O(n)$ time [BH01a], where n is the number of vertices of the convex-hull of K. The more powerful result showing that a convex body can be approximated by an ellipsoid (up to a scaling factor of d), is known as John's Theorem [Har11].

We need the following variant of the algorithm of Barequet and Har-Peled [BH01a].

Lemma 3.1. *Given a convex polygon* Z *in the plane, with* n *vertices, one can compute, in linear time, a rectangle* r_Z *and a point* $\vec{z}$*, such that* $\vec{z} + \mathsf{r}_\mathsf{Z} \subseteq \mathsf{Z} \subseteq \vec{z} + 5\mathsf{r}_\mathsf{Z}$.



Proof. This is all well known, and we include the details for the sake of completeness. Using rotating caliper [Tou83] compute the two vertices u and v of Z realizing its diameter. Let w be the vertex of Z furthest away from uv, Consider the rectangle r'_Z having its base on uv, having half the height of $\triangle uvw$, and contained inside this triangle. Now, let $\vec{z}$ be the center of r'_Z, and set $\mathsf{r}_\mathsf{Z} = \mathsf{r}'_\mathsf{Z} - \vec{z}$, see figure on the right. It is now easy to verify that the claim holds with r_Z and $\vec{z}$. ■

Observation 3.2. *Given two bodies* $\mathsf{X}, \mathsf{Y} \subseteq \mathbb{R}^2$ *and a non-singular affine transformation* M*, we have* $\dfrac{\mathrm{area}(\mathsf{X})}{\mathrm{area}(\mathsf{Y})} = \dfrac{\mathrm{area}(\mathsf{M}(\mathsf{X}))}{\mathrm{area}(\mathsf{M}(\mathsf{Y}))}$.

Since a similar construction is described by Ahn *et al.* [ACP$^+$07], we delegate the proof of this lemma to Appendix A.

Lemma 3.3 (approxPolygon). *Given a convex polygon* P*, and a parameter* $m > 0$*, we can compute, in* $O(|\mathsf{P}|)$ *time, a convex polygon* P' *with* $O(m)$ *vertices, such that (i)* $\mathsf{P}' \subseteq \mathsf{P}$*, and (ii) for any point* $\mathsf{p} \in \mathsf{P}$*, its distance from* P' *is at most* $\omega(\mathsf{P})/m$*, where* $\omega(\mathsf{P})$ *is the width of* P*.*

3.2. The level set of the area of overlap function

Definition 3.4. *The superlevel set of a function* $f : \mathbb{R}^d \to \mathbb{R}$*, for a value* α *is the set* $L_\alpha(f) = \left\{ \mathsf{p} \in \mathbb{R}^d \;\middle|\; f(\mathsf{p}) \geq \alpha \right\}$*. We will refer to it as the* **α-slice** *of* f*.*

Lemma 3.5. *Given two convex polygons* X *and* Y*, the slice* $\mathsf{Z} = L_\alpha(\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y}))$ *is convex, and has complexity* $O(m)$*, where* $m = |\mathsf{X}|\,|\mathsf{Y}|$*. Furthermore, given a point* $\mathsf{p} \in \mathsf{Z}$*, the convex body* Z *can be computed in* $O(m \log m)$ *time.*

Proof. Along any line, the function $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$ is unimodal [dBCD$^+$98]. This directly implies that on any segment joining two points of the boundary of an α-slice, the function will have values greater than α, and hence, it will lie inside the α-slice. Therefore, Z is convex.

As one translates Y over X, as long as the same pairs of edges intersect, the function governing the overlap function will remain the same quadratic polynomial – the function changes form whenever a vertex is being swept over.

Now, consider an edge e of X and an edge e' of Y. All the translations t for which e intersects $e' + \mathsf{t}$ map out a parallelogram $\pi(e, e')$, which has edges parallel to e and e'. The edges of these parallelograms correspond to translations, where some vertex of Y lies on an edge of X, or vice versa, and the vertices correspond to those translations where some vertex of X coincides with some vertex of Y.

So, consider the arrangement $\mathcal{A}$ of all lines passing through the edges of these parallelograms – it is defined by $O(|\mathsf{X}|\,|\mathsf{Y}|)$ lines, and as such has complexity $O\big(|\mathsf{X}|^2\,|\mathsf{Y}|^2\big)$ overall. By convexity, the boundary $\gamma = \partial\mathsf{Z}$ intersects every line at most twice, and hence the curve γ visits at most $O(|\mathsf{X}|\,|\mathsf{Y}|)$ faces of this arrangement (we count a visit to the same face with multiplicity). Thus, γ can be broken into $O(|\mathsf{X}|\,|\mathsf{Y}|)$ arcs, each one of them of constant descriptive complexity (i.e., its the boundary of a slice of a quadratic function, between intersection points of this curve with two lines).

As for the algorithm, set $\mathsf{t} = \mathsf{p}$, and compute the overlap function value at p by sweeping the two polygons, computing their intersection polygon, and then computing the vertical decomposition of this intersection polygon. It is easy to verify, that the overlap function is a quadratic function of t, and its the sum of the areas of the vertical trapezoids, each one of them can be decomposed into two triangles, and the coordinates of the vertices of each triangle are affine function of t. As such, the area of each triangle is a quadratic function, and adding them up give us the formula for the area of the whole overlap polygon. Now, as t moves, every time the vertical decomposition changes as a vertex of one polygon is swept over by the boundary of the other polygon, this corresponds to a local change in the vertical decomposition of the overlap polygon, and the overlap function can be changed in constant time. Thus, starting from p, we move t up, updating the overlap function as necessary till we reach a point, where the overlap function has value α. At that point, we trace out the outer zone of $\partial\mathsf{Z}$ – since this zone

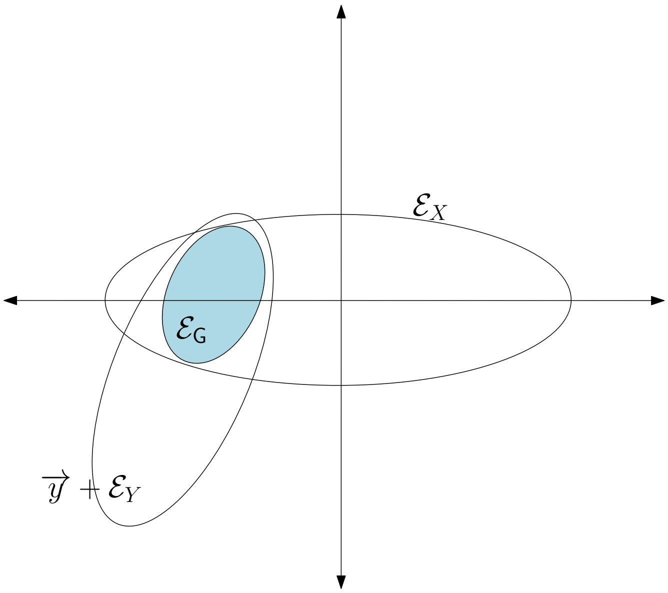


Figure 3.1:

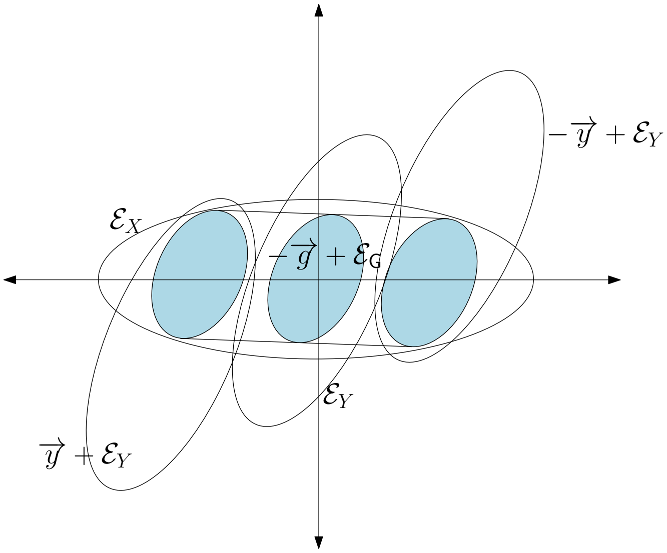

$-\overrightarrow{y} + \mathcal{E}_Y$
$\mathcal{E}_X$
$-\overrightarrow{g} + \mathcal{E}_\mathsf{G}$
$\mathcal{E}_Y$
$\overrightarrow{y} + \mathcal{E}_Y$

Figure 3.2:

is defined by $m = O(|\mathsf{X}|\,|\mathsf{Y}|)$ rays, this zone has complexity $O(m)$ (note, that the exact bound on the worst case complexity of the faces of the zone inside a convex region is still open, and it is between $O(m)$ and $O(m\alpha(m))$, where $\alpha(n)$ is the inverse Ackerman function). Specifically, we are walking on the arrangement of $\mathcal{A}$ starting on a point on a line that is on $\partial\mathsf{Z}$, go clockwise on the edges of the face till we encounter $\partial\mathsf{Z}$ again, move to the next face at this point, and repeat. To perform this, we need an efficient data-structure for doing a walk in a planar arrangement [Har00] – in this case since this is an arrangement of lines, we can use a data-structure for dynamic maintenance of convex-hull. In particular, Brodal and Jacob presented a data-structure [BJ02] that supports this kind of operations in $O(\log m)$ amortized time per update. As such, computing $\partial\mathsf{Z}$ takes $O(m \log m)$ time overall. ■

3.3. The shape of the polygon realizing the maximum area overlap

In the following, all the ellipses being considered are centered in the origin.

Lemma 3.6. *Given two ellipses $\mathcal{E}_1$ and $\mathcal{E}_2$, the translation which maximizes their area of overlap is the one in which their centers are the same points.*

Proof. Translate $\mathcal{E}_1$ and $\mathcal{E}_2$ such that their centers are at the origin. Consider any unit vector $\vec{u}$, translate $\mathcal{E}_2$ along the direction of $\vec{u}$, and consider the behavior of the overlap function $f(x) = \mu\Big(\mathcal{E}_1, \mathcal{E}_2 + x\vec{u}\Big)$, where x varies from $-\infty$ to $+\infty$. The function f is unimodal [dBCD+98]. By symmetry, we have

$$f(x) = \mu\Big(\mathcal{E}_1, \mathcal{E}_2 + x\vec{u}\Big) = \mu\Big(-\mathcal{E}_1,\, -(\mathcal{E}_2 + x\vec{u})\Big) = \mu\Big(\mathcal{E}_1, \mathcal{E}_2 - x\vec{u}\Big) = f(-x),$$

as $\mathcal{E}_i = -\mathcal{E}_i$. If the maximum is attained at $x \neq 0$, we will get another maximum at $-x$, which implies, as f unimodal, that $f(0) = f(x) = f(-x)$, as desired. ■

Lemma 3.7. *Consider two ellipses $\mathcal{E}_\mathsf{X}$ and $\mathcal{E}_\mathsf{Y}$ in the plane, and consider any two vectors $\vec{x}$ and $\vec{y}$, then there is a vector $\vec{u}$ such that $\vec{u} + (\vec{x} + \mathcal{E}_\mathsf{X}) \cap (\vec{y} + \mathcal{E}_\mathsf{Y}) \subseteq 2\mathcal{E}_\mathsf{X} \cap 2\mathcal{E}_\mathsf{Y}$.*

Proof. For the sake of simplicity of exposition, assume that $\vec{x} = 0$. Now, consider the intersection $\mathsf{G} = \mathcal{E}_{\mathsf{X}} \cap (\vec{y} + \mathcal{E}_{\mathsf{Y}})$, and let $\mathcal{E}_{\mathsf{G}}$ be the largest area ellipse contained inside G. John's theorem implies that there is a translation vector $\vec{g}$, such that $\vec{g} + \mathcal{E}_{\mathsf{G}} \subseteq \mathsf{G} \subseteq \vec{g} + 2\mathcal{E}_{\mathsf{G}}$, see Figure 3.1.

Observe that $\vec{g} + \mathcal{E}_{\mathsf{G}} \subseteq \mathcal{E}_{\mathsf{X}}$, and by the symmetry of $\mathcal{E}_{\mathsf{G}}$ and $\mathcal{E}_{\mathsf{X}}$, we have that $-\vec{g} + \mathcal{E}_{\mathsf{G}} = -\vec{g} - \mathcal{E}_{\mathsf{G}} \subseteq -\mathcal{E}_{\mathsf{X}} = \mathcal{E}_{\mathsf{X}}$. This by convexity implies that $\mathcal{E}_{\mathsf{G}} \subseteq \mathcal{E}_{\mathsf{X}}$. A similar argument implies that $\mathcal{E}_{\mathsf{G}} \subseteq \mathcal{E}_{\mathsf{Y}}$. As such, $\mathcal{E}_{\mathsf{G}} \subseteq \mathcal{E}_{\mathsf{X}} \cap \mathcal{E}_{\mathsf{Y}}$.

Thus, we have that $\mathsf{G} \subseteq \vec{g} + 2\mathcal{E}_{\mathsf{G}} \subseteq \vec{g} + 2\mathcal{E}_{\mathsf{X}} \cap 2\mathcal{E}_{\mathsf{Y}}$, as desired. ■

Lemma 3.8 ([Har11, Lemma 22.5]). *Any convex set $K \subseteq \mathbb{R}^d$ contained in a unit square, contains a ball of radius* $\mathrm{area}(K)/8$

The following lemma is one of our key insights – the maximum area of intersection of two polygons contains any intersection of translated copies of these polygons up to translation and a constant factor scaling.

Lemma 3.9. *Let X and Y be two convex polygons, and let M be the polygon realizing their maximum area of intersection under translation. Let $\vec{u}$ be any vector in the plane, and consider the polygon $\mathsf{D} = \mathsf{X} \cap (\vec{u} + \mathsf{Y})$, then there exists a vector $\vec{v}$ such that, $\vec{v} + \mathsf{D} \subseteq c_0\mathsf{M}$, for some fixed constant c_0.*

Proof. Let $\mathcal{E}_{\mathsf{X}}$ (resp., $\mathcal{E}_{\mathsf{Y}}$) denote the maximum area ellipse (centered at the origin) contained inside X (resp. Y). By John's Theorem, we have $\vec{x} + \mathcal{E}_{\mathsf{X}} \subseteq \mathsf{X} \subseteq \vec{x} + 2\mathcal{E}_{\mathsf{X}}$ and $\vec{y} + \mathcal{E}_{\mathsf{Y}} \subseteq \mathsf{Y} \subseteq \vec{y} + 2\mathcal{E}_{\mathsf{Y}}$, where $\vec{x}, \vec{y}$ are some vector. Let $\mathsf{B} = \mathcal{E}_{\mathsf{X}} \cap \mathcal{E}_{\mathsf{Y}}$, and let $\mathcal{E}_{\mathsf{B}}$ be the maximum area ellipse contained inside B. Observe that B is symmetric and centered at the origin, and by John's theorem $\mathcal{E}_{\mathsf{B}} \subseteq \mathsf{B} \subseteq 2\mathcal{E}_{\mathsf{B}}$.

By Lemma 3.7, there are vectors $\vec{z}$ and $\overrightarrow{w}$, such that

$$\begin{aligned}\mathsf{D} = \mathsf{X} \cap (\vec{u} + \mathsf{Y}) &\subseteq (\vec{x} + 2\mathcal{E}_{\mathsf{X}}) \cap (\vec{z} + \vec{y} + 2\mathcal{E}_{\mathsf{Y}}) \subseteq \overrightarrow{w} + 4\mathcal{E}_{\mathsf{X}} \cap 4\mathcal{E}_{\mathsf{Y}} = \overrightarrow{w} + 4\mathsf{B}\\ &\subseteq \overrightarrow{w} + 8\mathcal{E}_{\mathsf{B}}.\end{aligned}$$

Applying a similar argument, we have that $\mathsf{M} \subseteq \overrightarrow{m} + 8\mathcal{E}_{\mathsf{B}}$, for some vector $\overrightarrow{m}$.

Apply the linear transformation that maps $\mathcal{E}_{\mathsf{B}}$ to $\mathrm{disk}(1/16)$, where $\mathrm{disk}(r)$ denotes the disk of radius r centered at the origin. By Observation 3.2, we can continue our discussion in the transformed coordinates. This implies that $\mathsf{M} - \overrightarrow{m} \subseteq \mathrm{disk}(1/2)$ (which is contained inside a unit square). By Lemma 3.8, there is a vector $\vec{x_1}$, such that $\vec{x_1} + \mathrm{disk}(\mathrm{area}(\mathsf{M})/8) \subseteq \mathsf{M}$.

Observe that $\mathsf{B} = \mathcal{E}_{\mathsf{X}} \cap \mathcal{E}_{\mathsf{Y}} \subseteq (-\vec{x} + \mathsf{X}) \cap (-\vec{y} + \mathsf{Y})$. As such, the area of B must be smaller than the area of M (by the definition of M). We thus have $\mathrm{area}(\mathsf{M}) \geq \mathrm{area}(\mathsf{B}) \geq \mathrm{area}(\mathcal{E}_{\mathsf{B}}) = \mathrm{area}(\mathrm{disk}(1/16))$ which is a constant bounded away from zero. Therefore,

$$\begin{aligned}\mathsf{D} &\subseteq \overrightarrow{w} + 8\mathcal{E}_{\mathsf{B}} = \overrightarrow{w} + \mathrm{disk}\left(\frac{1}{2}\right) = \overrightarrow{w} + \frac{4}{\mathrm{area}(\mathsf{M})} \cdot \mathrm{disk}\left(\frac{\mathrm{area}(\mathsf{M})}{8}\right)\\ &\subseteq \overrightarrow{w} + \frac{4}{\mathrm{area}(\mathsf{M})}(\mathsf{M} - \vec{x_1}),\end{aligned}$$

which implies the claim. ■

3.3.1. Constant approximation to the maximum overlap

Lemma 3.10 (constApproxByRect). *Let X and Y be two convex polygons, and let M be the polygon realizing their maximum area intersection under translation. Then, one can compute, in $O(|\mathsf{X}| + |\mathsf{Y}|)$ time, a rectangle r, such that $\mathsf{r} \subseteq \vec{u} + \mathsf{M} \subseteq c_r\mathsf{r}$, where c_r is a constant. That is, one can compute a constant factor approximation to the maximum area overlap in linear time.*

Furthermore, for any translation t_{Y}, we have that $\mathsf{X} \cap (\mathsf{Y} + \mathsf{t}_{\mathsf{Y}}) \sqsubseteq c_r\mathsf{r}$.

Proof. We are going to implement the algorithmic proof of Lemma 3.9. Instead of John's ellipsoid we use the rectangle of Lemma 3.1. Clearly, the proof of Lemma 3.9 goes through with the constants being somewhat worse. Specifically, we compute, in linear time, vectors $\vec{x}, \vec{y}$, and rectangles $\mathsf{r}_\mathsf{X}, \mathsf{r}_\mathsf{Y}$, such that $\vec{x} + \mathsf{r}_\mathsf{X} \subseteq \mathsf{X} \subseteq \vec{x} + 5\mathsf{r}_\mathsf{X}$ and $\vec{y} + \mathsf{r}_\mathsf{Y} \subseteq \mathsf{Y} \subseteq \vec{y} + 5\mathsf{r}_\mathsf{Y}$. Again, compute a rectangle r_M, such that $\mathsf{r}_\mathsf{M}/5 \subseteq \mathsf{r}_\mathsf{X} \cap \mathsf{r}_\mathsf{Y} \subseteq \mathsf{r}_\mathsf{M}$. Arguing as in Lemma 3.9, and setting $\mathsf{r} = \mathsf{r}_\mathsf{M}/c_3$, for some constant c_3, is the desired rectangle. ■

4. Approximating the overlap function of convex polygons

Definition 4.1. *Given two convex polygons* X *and* Y *in the plane, of total complexity* n*, and parameters* $\varepsilon \in (0,1)$, ν, ρ*, a function* $\psi(\mathsf{t})$ *is* **(ε, ν, ρ)-approximation** *of* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*, if the following conditions hold:*

(A) $\forall \mathsf{t} \in \mathbb{R}^2$*, we have* $|\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y}) - \psi(\mathsf{t})| \leq \varepsilon \mu_{\max}(\mathsf{X}, \mathsf{Y})$.

(B) There are convex polygons $\mathsf{P}_1, \ldots, \mathsf{P}_\nu$*, each of maximum complexity* ρ*, such that inside every face of the arrangement* $\mathcal{A} = \mathcal{A}(\mathsf{P}_1, \ldots, \mathsf{P}_\nu)$*, the approximation function* $\psi(\mathsf{t})$ *is the same quadratic function.*

That is, the total descriptive complexity of $\psi(\cdot)$ *is the complexity of the arrangement* $\mathcal{A}$.

Algorithm 4.2. *The input is two convex polygons* X *and* Y *in the plane, of total complexity* n*, and a parameter* $\varepsilon \in (0,1)$*. As a first step, the algorithm is going to approximate* X *and* Y *as follows:*

(A) $\mathsf{r}_\mathsf{M} \leftarrow$ **constApproxByRect**(X, Y)*, see Lemma 3.10.*

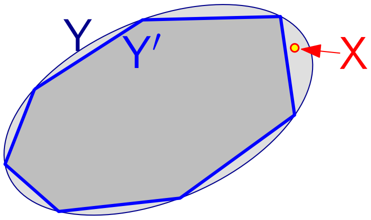
(B) $\mathcal{T} \leftarrow$ *affine transformation that maps* $2c_r\mathsf{r}_\mathsf{M}$ *to* $[0,1]^2$.

(C) $\mathsf{X}'_\mathcal{T} \leftarrow$ **approxPolygon**$(\mathcal{T}(\mathsf{X}), N)$ *and* $\mathsf{Y}'_\mathcal{T} \leftarrow$ **approxPolygon**$(\mathcal{T}(\mathsf{Y}), N)$.
See Lemma 3.3, here $N = \lceil c_4/\varepsilon \rceil$*, and* c_4 *is a sufficiently large constant.*

(D) $\mathsf{X}' \leftarrow \mathcal{T}^{-1}(\mathsf{X}'_\mathcal{T})$ *and* $\mathsf{Y}' \leftarrow \mathcal{T}^{-1}(\mathsf{Y}'_\mathcal{T})$.

4.1. If one polygon is smaller than the other

Assume, without loss of generality, that X is smaller than Y, that is, X can be translated so that it is entirely contained inside Y (i.e., $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \leq 1$, see Lemma 2.3). The maximum area of overlap is now equal to $\mathrm{area}(\mathsf{X})$. The challenge is, that for any approximation of Y, we can always have a sufficiently small X which can be placed in $\mathsf{Y} \setminus \mathsf{Y}'$, as shown in the figure on the right. Therefore for all those translations for which X is placed inside $\mathsf{Y} \setminus \mathsf{Y}'$, our approximation will show zero overlap, even though the actual overlap is $\mathrm{area}(\mathsf{X})$.



To get around this problem, we will first approximate the smaller polygon X, using our approximation scheme, to get polygon X', then we will compute level sets of the overlap function and use them to approximate it.

Lemma 4.3. *Given convex polygons* X *and* Y*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) < 1$*, and parameter* $\varepsilon > 0$*, and let* X' *be the approximation to* X*, as computed by Algorithm 4.2. Then, we have, for all translations* $\mathsf{t} \in \mathbb{R}^2$*, that* $\left| \mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y}) \right| \leq \varepsilon \mu_{\max}(\mathsf{X}, \mathsf{Y})$.

Proof. Consider the overlap of $\mathsf{X}_\mathcal{T} = \mathcal{T}(\mathsf{X})$ and $\mathsf{Y}_\mathcal{T} = \mathcal{T}(\mathsf{Y})$. Lemma 3.10 implies that any intersection polygon of $\mathsf{X}_\mathcal{T}$ and $\mathsf{Y}_\mathcal{T}$ can be contained (via translation) in $\mathcal{T}(c_r\mathsf{r}_\mathsf{M})$ (which is a translation of the square $[0,1/2]^2$). Clearly, in this case, $\mathsf{X}_\mathcal{T}$ and $\mathsf{X}'_\mathcal{T}$ can both be translated to be contained in this square, both contain a disk of constant radius, the maximum distance between $\mathsf{X}_{\mathcal{T}}$ and $\mathsf{X}'_{\mathcal{T}}$ is $O(\varepsilon)$, and the total area of $\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}}$ is $O(\varepsilon)$, as the perimeter of $\mathsf{X}_{\mathcal{T}} \leq 4$. Thus, setting c_4 to be sufficiently large, implies that $\mathrm{area}(\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}}) \leq \varepsilon \mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}})$, as $\mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}}) = \Omega(1)$. This implies that $\left|\mu(\mathsf{X}'_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}}) - \mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}})\right| \leq \varepsilon \mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}})$, which implies the claims by applying $\mathcal{T}^{-1}$ to both sides. ■

Therefore, $\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y})$ is a good approximation for $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$. However, $\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y})$ has complexity $O\left(|\mathsf{X}'|^2 |\mathsf{Y}|^2\right)$ [dBCD+98], in the worst case, which is still too high.

Lemma 4.4 (approxLevelSet). *Given two convex polygons* X *and* Y*, of total complexity* n*, and a parameter* ε*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) < 1$*, then one can construct in* $O\left(n/\varepsilon^2\right)$ *time, a* $(\varepsilon, O(1/\varepsilon^2), O(n/\varepsilon^2))$*-approximation* $\psi(\cdot)$ *to* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*.*

Proof. There is a translation of X such that it is contained completely in Y. Approximate X from the outside by a rectangle r, using Lemma 3.1. Next, spread a grid in r by partitioning each of its edges into $O(1/\varepsilon)$ equal length intervals. Let S be the set of points of the grid that are in X. It is easy to verify, that for any convex body Z and a translation t, we have

$$\left|\mu(\mathsf{X}, \mathsf{t} + \mathsf{Z}) - \frac{|(\mathsf{t} + \mathsf{Z}) \cap \mathsf{S}|}{|\mathsf{S}|}\right| \leq \varepsilon \, \mathrm{area}(\mathsf{X})$$

Namely, to approximate the overlap area for $\mathsf{t} + \mathsf{Y}$, we need to count the number of points of S that it covers. To this end, for each point $\mathsf{p} \in \mathsf{S}$, we generate a 180° rotated and translated copy of Y, denoted by Y'_{p}, such that $\mathsf{p} \in \mathsf{t} + \mathsf{Y}$ if and only if $\mathsf{t} \in \mathsf{Y}'_{\mathsf{p}}$.

Clearly, the generated set of polygons is the desired $(\varepsilon, O(1/\varepsilon^2), O(n/\varepsilon^2))$-approximation $\psi(\cdot)$ to $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$.

The time to build this approximation is $O(n/\varepsilon^2)$. ■

We next describe a slightly slower algorithm that generates a slightly better approximation.

Lemma 4.5 (approxLevelSet). *Given two convex polygons* X *and* Y*, of total complexity* n*, and a parameter* ε*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) < 1$*, then one can construct in* $O\left(\varepsilon^{-2} n \log n\right)$ *time, a* $(\varepsilon, O(1/\varepsilon), O(n/\varepsilon))$*-approximation* $\psi(\cdot)$ *to* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*.*

Proof. We compute X', as above with approximation parameter $\epsilon = \varepsilon/4$. Next, using the algorithm of [dBCD+98], we compute $\mu_{\max}(\mathsf{X}', \mathsf{Y})$ in $O(n' \log n') = O(n \log n)$ time, where $n' = |\mathsf{X}'| + |\mathsf{Y}| = O(n)$.

We approximate the function $\mu(\mathsf{X}', \mathsf{Y} + \mathsf{t})$ by constructing the α_i-slices, where $\alpha_i = \min(1, (i + 1)\epsilon)\mu_{\max}(\mathsf{X}', \mathsf{Y})$, for $i = 0, 1, \ldots, M$, where $M = \lceil 1/\epsilon \rceil$. To this end, we deploy the algorithm of Lemma 3.5 for each slice, which takes $O(m \log m)$ time, where $m = |\mathsf{X}'| \, |\mathsf{Y}| = O(n/\varepsilon)$.

Let us denote the ith region constructed, by S_i, that is, S_i is a convex figure whose boundary corresponds to all translations t such that $\mu(\mathsf{X}', \mathsf{Y} + \mathsf{t}) = \alpha_i$. Clearly, S_{i+1} lies entirely within S_i. Given the description of S_i's, for any translation t, we define

$$\psi(\mathsf{t}) = \begin{cases} \alpha_i & \mathsf{t} \in \mathsf{S}_i \setminus \mathsf{S}_{i+1} \\ 0 & \mathsf{t} \notin \mathsf{S}_0. \end{cases}$$

It is now straightforward to verify this is the desired approximation. Indeed, for $\mathsf{t} \notin \mathsf{S}_0$, we have by Lemma 4.3, that

$$\begin{aligned}\left|\psi(\mathsf{t}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\right| &\leq \left|\psi(\mathsf{t}) - \mu(\mathsf{X}', \mathsf{t} + \mathsf{Y})\right| + \left|\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\right| \leq \alpha_0 + \epsilon \mu_{\max}(\mathsf{X}, \mathsf{Y}) \\ &= \epsilon(\mu_{\max}(\mathsf{X}', \mathsf{Y}) + \mu_{\max}(\mathsf{X}, \mathsf{Y})) \leq 2\epsilon \mu_{\max}(\mathsf{X}, \mathsf{Y}) = (\varepsilon/2)\mu_{\max}(\mathsf{X}, \mathsf{Y}),\end{aligned}$$

as desired. Similarly, if $\mathsf{t} \in \mathsf{S}_i \setminus \mathsf{S}_{i+1}$ then $\alpha_i \leq \mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}) \leq \alpha_{i+1}$, and

$$\begin{aligned}\Big|\mu'(\mathsf{t}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\Big| &\leq \Big|\mu'(\mathsf{t}) - \mu(\mathsf{X}', \mathsf{t} + \mathsf{Y})\Big| + \Big|\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\Big| \\ &\leq \alpha_{i+1} - \alpha_i + \epsilon\mu_{\max}(\mathsf{X}, \mathsf{Y}) = \epsilon\mu_{\max}(\mathsf{X}', \mathsf{Y}) + \epsilon\mu_{\max}(\mathsf{X}, \mathsf{Y}) \\ &\leq (\varepsilon/2)\mu_{\max}(\mathsf{X}, \mathsf{Y}) .\end{aligned}$$

Clearly, this function is defined by an onion like set of $\nu = O(1/\varepsilon)$ polygons, and the maximum complexity of these polygons is $\rho = O(m) = O(n/\varepsilon)$.

The overall running time is dominated by computing the slices, which takes overall $O(\nu m \log m) = O(n\varepsilon^{-2} \log n\varepsilon^{-1})$ time. ■

4.2. If the two polygons are incomparable

The more interesting case, is when the maximum intersection of X and Y is significantly smaller than both polygons; that is, $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \geq 1$ and $\mathrm{ssim}(\mathsf{Y}, \mathsf{X}) \geq 1$. Surprisingly, in this case, we can approximate both polygons simultaneously.

Lemma 4.6. *Given convex polygons* X *and* Y*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \geq 1$ *and* $\mathrm{ssim}(\mathsf{Y}, \mathsf{X}) \geq 1$*, then the widths of* $\mathsf{X}_{\mathcal{T}} = \mathcal{T}(\mathsf{X})$ *and* $\mathsf{Y}_{\mathcal{T}} = \mathcal{T}(\mathsf{Y})$*, as computed by Algorithm 4.2, are bounded by* 7*.*

Proof. Let $\omega_{\mathsf{X}} = \mathrm{width}(\mathsf{X}_{\mathcal{T}})$ and $\omega_{\mathsf{Y}} = \mathrm{width}(\mathcal{T}(\mathsf{Y}))$. By Lemma 2.2, we have that $\mathrm{disk}(\omega_{\mathsf{X}}/3.5) \sqsubseteq \mathsf{X}_{\mathcal{T}}$ and $\mathrm{disk}(\omega_{\mathsf{Y}}/3.5) \sqsubseteq \mathsf{Y}_{\mathcal{T}}$.

So, assume for the sake of contradiction, that $\omega_{\mathsf{X}} \geq 7$. This implies that $\mathsf{X}_{\mathcal{T}}$ contains a disk of radius 2, which in turn contains the unit square. In particular, let $\ell = \mathrm{diameter}(\mathsf{Y}_{\mathcal{T}})$, if $\ell < 2$, then $\mathsf{Y}_{\mathcal{T}}$ is contained in a disk of radius 2, implies that $\mathsf{Y}_{\mathcal{T}} \sqsubseteq \mathsf{X}_{\mathcal{T}}$, a contradiction.

Otherwise, if $\ell > 2$ then there is a translation of $\mathsf{Y}_{\mathcal{T}}$ such that its intersection with $\mathsf{Y}_{\mathcal{T}}$ has length > 2 (indeed, consider the segment realizing the diameter of $\mathsf{Y}_{\mathcal{T}}$, and translate it so its middle point is in the center of the disk of radius 2 inside $\mathsf{X}_{\mathcal{T}}$). But then, this intersection is not contained in $[0, 1]^2$ under any translation, which contradicts Lemma 3.10.

The case $\omega_{\mathsf{Y}} \geq 7$ is handled in a similar fashion. ■

Lemma 4.7. *Given two convex polygons* X *and* Y*, of total complexity* n*, and a parameter* ε*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \geq 1$ *and* $\mathrm{ssim}(\mathsf{Y}, \mathsf{X}) \geq 1$*, then one can construct in* $O\big(n + 1/\varepsilon^2\big)$ *time, a* $(\varepsilon, O(1/\varepsilon), O(1/\varepsilon))$*-approximation* $\psi(\cdot)$ *to* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*.*

Proof. Use Algorithm 4.2 to compute X' and Y', both of complexity $O(1/\varepsilon)$. We set $\psi(\mathsf{t}) = \mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}')$. [dBCD+98] describes how to describe the function $\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}')$ as an arrangement of $O(|\mathsf{X}'| + |\mathsf{Y}'|)$ polygons, each of complexity $|\mathsf{X}'|$ or $|\mathsf{Y}'|$. Thus, this result in the desired approximation, that has total complexity $O(1/\varepsilon^2)$, and that can be computed in $O(n + 1/\varepsilon^2)$ time.

In the following, we use the notation of Algorithm 4.2: $\mathsf{X}_{\mathcal{T}} = \mathcal{T}(\mathsf{X})$ and $\mathsf{Y}_{\mathcal{T}} = \mathcal{T}(\mathsf{Y})$, $\mathsf{X}'_{\mathcal{T}} =$ **approxPolygon**$(\mathcal{T}(\mathsf{X}), N)$, and $\mathsf{Y}'_{\mathcal{T}} =$ **approxPolygon**$(\mathcal{T}(\mathsf{X}), N)$.

Lemma 3.10 implies that any intersection polygon of $\mathsf{X}_{\mathcal{T}}$ and $\mathsf{Y}_{\mathcal{T}}$ can be contained in $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$. The error due to approximation of $\mathsf{X}_{\mathcal{T}}$, is $\mathrm{area}(\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}})$. The part of this error that can contribute to the area of overlap, is bounded by portion of $\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}}$, which can be included inside $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$.

The length of the boundary of $\mathsf{X}'_{\mathcal{T}}$, which can be placed inside $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$, is bounded by the perimeter of $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$. Also, our approximation scheme (Lemma 3.3) ensures that the distance between $\mathsf{X}_{\mathcal{T}}$ and $\mathsf{X}'_{\mathcal{T}}$ (along the direction of shortest diameter) is bounded by $\omega(\mathsf{X}_{\mathcal{T}})\,/N$, where N is a chosen parameter.

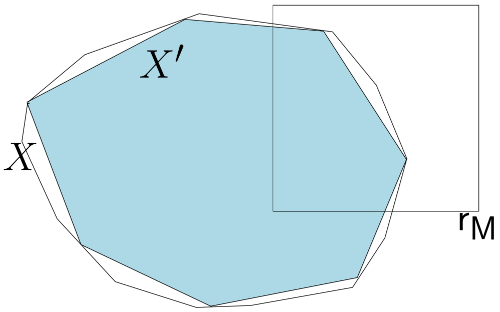


Figure 4.1: Error bound.

Consider any fixed t – for the sake of simplicity of exposition, we assume the intersection $\mathsf{X}_{\mathcal{T}} \cap (\mathsf{t} + \mathsf{Y}_{\mathcal{T}})$ as being inside the square $\mathcal{T}(c_r \mathsf{r_M})$ (if not, we can translate $\mathcal{T}(c_r \mathsf{r_M})$ so this assumption holds). As $N = \lceil c_4/\varepsilon \rceil$, and for c_4 sufficiently large constant, the error due to the approximation of $\mathsf{X}_{\mathcal{T}}$ is bounded by

$$\begin{aligned}\mathcal{E}_{\mathsf{X}} &= \Big|\mu(\mathsf{X}'_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}}) - \mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}})\Big| = \Big|\mu(\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}})\Big| \\ &\leq \text{perimeter}\Big(\mathsf{X}_{\mathcal{T}} \cap (\mathsf{t} + \mathsf{Y}_{\mathcal{T}})\Big) \times \frac{\omega(\mathsf{X}_{\mathcal{T}})}{N} \leq \text{Perimeter}(\mathcal{T}(c_r \mathsf{r_M})) \times \frac{\omega(\mathsf{X}_{\mathcal{T}})}{N} \\ &\leq 4 \times \frac{7}{N} \leq \frac{\varepsilon}{4} \mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}}),\end{aligned}$$

by Lemma 4.6, and since by Lemma 3.10 we have $\mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}}) = \Omega(1)$. A symmetric argument works for the error $\mathcal{E}_{\mathsf{Y}}$ of the overlap caused by the approximation of $\mathsf{Y}_{\mathcal{T}}$ by $\mathsf{Y}'_{\mathcal{T}}$. We conclude that

$$\begin{aligned}\mathcal{E}_{\mathcal{T}} &= \Big|\mu(\mathsf{X}'_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}'_{\mathcal{T}}) - \mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}})\Big| \\ &\leq \Big|\mu(\mathsf{X}'_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}'_{\mathcal{T}}) - \mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}'_{\mathcal{T}})\Big| + \Big|\mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}'_{\mathcal{T}}) - \mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}})\Big| \\ &\leq \Big|\mu(\mathsf{X}'_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}}) - \mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}})\Big| + \mathcal{E}_{\mathsf{Y}} \leq \mathcal{E}_{\mathsf{X}} + \mathcal{E}_{\mathsf{Y}} \leq \frac{\varepsilon}{2} \mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}}),\end{aligned}$$

since $\mathsf{Y}'_{\mathcal{T}} \subseteq \mathsf{Y}_{\mathcal{T}}$. By applying $\mathcal{T}^{-1}$ to the above, we get that for any translation t, it holds

$$\mathcal{E} = \Big|\psi(\mathsf{t}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\Big| = \Big|\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}') - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\Big| \leq \frac{\varepsilon}{2} \mu_{\max}(\mathsf{X}, \mathsf{Y}). \qquad \blacksquare$$

4.2.1. The result.

By combining Lemma 4.4 and Lemma 4.7 (deciding which one to apply can be done by computing $\text{ssim}(\mathsf{X}, \mathsf{Y})$ and $\text{ssim}(\mathsf{Y}, \mathsf{X})$, which takes $O(n)$ time), we get the following.

Lemma 4.8. *Given two convex polygons X and Y, of total complexity n, and a parameter ε, one can construct in $O\big(n/\varepsilon^2\big)$ time, a $(\varepsilon, O(1/\varepsilon^2), O(n/\varepsilon^2))$-approximation $\psi(\cdot)$ to $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$.*

5. Approximating the Maximum Overlap of Polygons

The input is two polygons P and Q in the plane, of total complexity n, each of them can be decomposed into at most k convex polygons. Our purpose is to find the translation that maximizes the area of overlap.

The Algorithm. We decompose the polygons P and Q into minimum number of interior disjoint convex polygons [KS02], in time $O(n + k^2 \min(k^2, n))$ (some of these convex polygons can be empty). Then, for every pair P_i, Q_j, we compute an $(\epsilon, O(1/\epsilon^2), O(n/\epsilon^2))$-approximation ψ_{ij} to the overlap function of P_i and Q_j, using Lemma 4.8, where $\epsilon = \varepsilon/k^2$.

Next, as each function ψ_{ij} is defined by an arrangement defined by $O(1/\epsilon^2)$ polygons, we overlay all these arrangements together, and compute for each face of the arrangement the function $\psi = \sum_{i,j} \psi_{ij}$. Inside such a face this function is the same, and it is a quadratic function. We then find the global maximum of this function, and return it as the desired approximation.

Analysis – Quality of approximation. For any translation t, we have that

$$\begin{aligned}\left|\mu(\mathsf{P}, \mathsf{t} + \mathsf{Q}) - \psi(\mathsf{t})\right| &\leq \sum_{i=1}^{k}\sum_{j=1}^{k} \left|\mu(\mathsf{P}_i, \mathsf{t} + \mathsf{Q}_j) - \psi_{ij}(\mathsf{t})\right| \leq \sum_{i=1}^{k}\sum_{j=1}^{k} \epsilon\mu_{\max}(\mathsf{P}_i, \mathsf{Q}_j) \leq \epsilon k^2 \mu_{\max}(\mathsf{P}, \mathsf{Q}) \\ &\leq \varepsilon\mu_{\max}(\mathsf{P}, \mathsf{Q}) .\end{aligned}$$

Analysis – Running time. Computing each of the k^2 approximation function, takes $O\big((k/\varepsilon)^2 n\big)$ time. Each one of them is a $(\varepsilon/k, O(k^2/\varepsilon^2), O(k^2 n/\varepsilon^2))$-approximation, which means that the final arrangement is the overlay of $O(k^4/\varepsilon^2)$ convex polygons, each of complexity $O(k^2 n/\varepsilon^2)$. In particular, any pair of such polygons can have at most $O(k^2 n/\varepsilon^2)$ intersection points, and thus the overall complexity of the arrangement of these polygons is $N = O\Big(\left(k^4/\varepsilon^2\right)^2 \left(k^2 n/\varepsilon^2\right)\Big) = O(k^{10}\varepsilon^{-6} n)$. Computing this arrangement can be done by a standard sweeping algorithm. Observing that every vertical line crosses only $O(k^4/\varepsilon^2)$ segments, imply that the sweeping can be done in $O(\log(k/\varepsilon))$ time per operation, which implies that the overall running time is

$$O\left(k^2 \frac{k^2}{\varepsilon^2} n + N \log \frac{k}{\varepsilon}\right) = O\left(\frac{k^{10}}{\varepsilon^6} n \log \frac{k}{\varepsilon}\right).$$

The result.

Theorem 5.1. *Given two simple polygons P and Q of total complexity n, one can compute a translation which ε-approximates the maximum area of overlap of P and Q. The time required is $O(c'n)$ where $c' = \dfrac{k^1 0}{\varepsilon^6} \log \dfrac{k}{\varepsilon}$, where k is the minimum number of convex polygons in the decomposition of P and Q.*

More specifically, one gets a data-structure, such that for any query translation t, one can compute, in $O(\log n)$ time, an approximation $\psi(\mathsf{t})$, such that $|\psi(\mathsf{t}) - \mu(\mathsf{P}, \mathsf{Q})| \leq \varepsilon\mu_{\max}(\mathsf{P}, \mathsf{Q})$, where $\mu_{\max}(\mathsf{P}, \mathsf{Q})$ is the maximum area of overlap between P and Q.

Note, that our analysis is far from tight. Specifically, for the sake of simplicity of exposition, it is loose in several places as far as the dependency on k and ε.

Acknowledgments

The authors would like to thank the anonymous referees for their insightful comments. In particular, the improved construction of Lemma 4.4 was suggested by an anonymous referee.

References

[ABT^{+}96] D. Avis, P. Bose, G. T. Toussaint, T. C. Shermer, B. Zhu, and J. Snoeyink. On the sectional area of convex polytopes. In *Proc. 12th Annu. Sympos. Comput. Geom.* (SoCG), pages 411–412, 1996.

[ACKY14] H.-K. Ahn, S.-W. Cheng, H. J. Kweon, and J. Yon. Overlap of convex polytopes under rigid motion. *Comput. Geom. Theory Appl.*, 47(1):15–24, 2014.

[ACP^{+}07] H.-K. Ahn, O. Cheong, C.-D. Park, C.-S. Shin, and A. Vigneron. Maximizing the overlap of two planar convex sets under rigid motions. *Comput. Geom. Theory Appl.*, 37(1):3–15, 2007.

[ACR13] H.-K. Ahn, S.-W. Cheng, and I. Reinbacher. Maximum overlap of convex polytopes under translation. *Comput. Geom. Theory Appl.*, 46(5):552–565, 2013.

[AFRW98] H. Alt, U. Fuchs, G. Rote, and G. Weber. Matching convex shapes with respect to the symmetric difference. *Algorithmica*, 21:89–103, 1998.

[AG00] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153. Elsevier, 2000.

[BH01a] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38:91–109, 2001.

[BH01b] G. Barequet and S. Har-Peled. Polygon containment and translational min-hausdorff-distance between segment sets are 3sum-hard. *Internat. J. Comput. Geom. Appl.*, 11(4):465–474, 2001.

[BJ02] G. S. Brodal and R. Jacob. Dynamic planar convex hull. In *Proc. 43th Annu. IEEE Sympos. Found. Comput. Sci.* (FOCS), pages 617–626, 2002.

[CEH07] O. Cheong, A. Efrat, and S. Har-Peled. On finding a guard that sees most and a shop that sells most. *Discrete Comput. Geom.*, 37(4):545–563, 2007.

[CL13] S.-W. Cheng and C.-K. Lam. Shape matching under rigid motion. *Comput. Geom. Theory Appl.*, 46(6):591–603, 2013.

[CLM05] B. Chazelle, D. Liu, and A. Magen. Sublinear geometric algorithms. *SIAM J. Comput.*, 35(3):627–646, 2005.

[dBCD^{+}98] M. de Berg, O. Cheong, O. Devillers, M. van Kreveld, and M. Teillaud. Computing the maximum overlap of two convex polygons under translations. *Theo. Comp. Sci.*, 31:613–628, 1998.

[dBCG^{+}04] M. de Berg, S. Cabello, P. Giannopoulos, C. Knauer, R. van Oostrum, and R. C. Veltkamp. Maximizing the area of overlap of two unions of disks under rigid motion. In *Scandinavian Workshop on Algorithms*, volume 3111 of *LNCS*, pages 138–149, 2004.

[GK92] P. Gritzmann and V. Klee. Inner and outer j-radii of convex bodies in finite-dimensional normed spaces. *Discrete Comput. Geom.*, 7:255–280, 1992.

[Har00] S. Har-Peled. Taking a walk in a planar arrangement. *SIAM J. Comput.*, 30(4):1341–1367, 2000.

[Har11] S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Mathematical Surveys and Monographs*. Amer. Math. Soc., 2011.

[KS02] J. M. Keil and J. Snoeyink. On the time bound for convex decomposition of simple polygons. *Internat. J. Comput. Geom. Appl.*, 12(3):181–192, 2002.

[MSW96] D. M. Mount, R. Silverman, and A. Y. Wu. On the area of overlap of translated polygons. *Computer Vision and Image Understanding: CVIU*, 64(1):53–61, 1996.

[ST94] M. Sharir and S. Toledo. Extremal polygon containment problems. *Comput. Geom. Theory Appl.*, 4:99–118, 1994.

[Tou83] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.

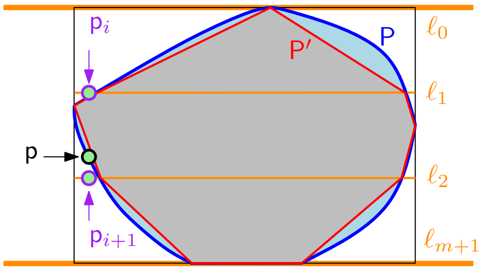
[Vig14] A. Vigneron. Geometric optimization and sums of algebraic functions. *ACM Trans. Algo.*, 10(1):4:1–4:20, 2014.

A. Proof of Lemma 3.3

Lemma 3.3 follows by the work of Ahn *et al.* [ACP+07]. For the sake of completeness we include the details here (our construction is somewhat different).

Proof. The width of P is realized by two parallel lines ℓ_0 and ℓ_{m+1} at a distance of at most $\omega(\mathsf{P})$ from each other. These two lines can be computed in linear time using rotating caliper [Tou83] (and one of them must pass through an edge of P). Similarly, it is easy to compute the two extreme points of P in the direction of ℓ. Adding the supporting lines to P through these points that are orthogonal to ℓ, results in a rectangle, that has five vertices of P on its boundary, and we mark these vertices. Next, we slice this rectangle by m parallel lines $\ell_1, \ldots, \ell_m$ into $m+1$ slices of the same width. We mark all the intersections of these lines with P. Next, let P' be the polygon formed by the convex-hull of all the marked points. It is easy to verify that P' can be computed in linear time.

Clearly, $\mathsf{P}' \subseteq \mathsf{P}$. For any point $\mathsf{p} \in \mathsf{P}$, consider the two lines ℓ_i and ℓ_{i+1} that p is contained between them. Consider the projection of p into ℓ_i and ℓ_{i+1}, denoted by p_i and p_{i+1} respectively. If any of these two projections are inside P, then we are done, as this portion of P is inside P', and the distance of projection is at most $\omega(\mathsf{P})/m$. Thus, the only remaining possibility is that both projections are outside P. But this implies that P's extreme point in the parallel direction to ℓ (or its reverse) must be in p's slice. In particular, the segment $\mathsf{p}_i\mathsf{p}_{i+1}$ (that includes p) must intersect P, and furthermore, since the aforementioned extreme point is a vertex of P', it follows that this segments must intersect P', implying the claim. ■



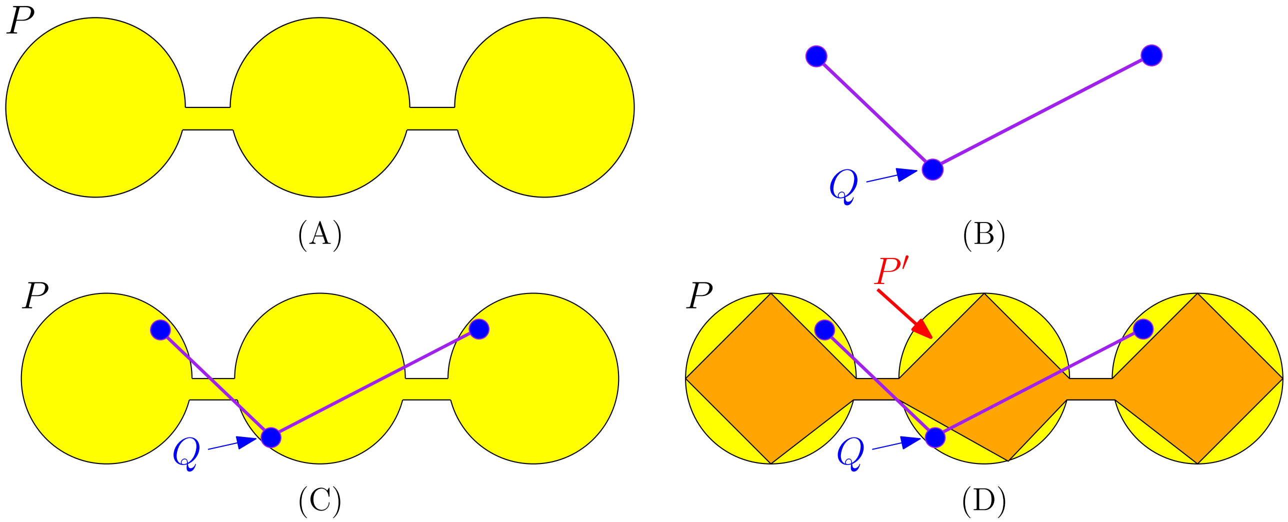


Figure B.1: Counterexample to naive approach for approximating overlap. (A), (B) Input polygons. (C) Optimal placement. (D) Approximate polygons fails to preserve optimal solution.

B. Counter example for some natural approaches

> "I have not failed. I've just found 10,000 ways that won't work." – Thomas Edison

Here we present a counterexample to the natural for our problem: Decompose the two polygons into their convex parts, approximate the parts, and then find the maximum overlap for the two reconstituted polygons.

So, consider the polygon P, depicted in Figure B.1 (A), and its arch-nemesis Q in Figure B.1 (B). The optimal overlap placement is depicted in Figure B.1 (C).

But any naive piecewise approximation of P by pieces will fail, because it creates cavities that might contain the whole overlap. For example, one possible solution for the approximate polygon P' and the maximum overlap, is depicted in Figure B.1 (D).

In particular, any placement that makes the approximate polygon P' cover two of the disks of Q, would fail to cover the third disk of Q. so, in this case, the best approximation suggested by the reviewer is (roughly) 2/3. This can be made to be arbitrarily bad by enlarging k.

Another approach that does not work. Another natural idea is to pick from each pair of polygons P_i and Q_j a set of directions such that if you approximate each pair with these directions than the pair overlap is approximated correctly. Note, however, that if the polygons are of completely different sizes then, inherently, you need unbounded number of directions if the maximum overlap of the original (non-convex polygons) align the pieces far from their piecewise overlap maximum.

On the Complexity of Clustered-Level Planarity and T-Level Planarity *

Patrizio Angelini[†], Giordano Da Lozzo[†], Giuseppe Di Battista[†],
Fabrizio Frati[◇], Vincenzo Roselli[†]

[†] Department of Engineering, Roma Tre University, Italy
{angelini,dalozzo,gdb,roselli}@dia.uniroma3.it
[◇] School of Information Technologies, The University of Sydney, Australia
fabrizio.frati@sydney.edu.au

Abstract. In this paper we study two problems related to the drawing of level graphs, that is, T-LEVEL PLANARITY and CLUSTERED-LEVEL PLANARITY. We show that both problems are $\mathcal{NP}$-complete in the general case and that they become polynomial-time solvable when restricted to proper instances.

1 Introduction and Overview

A level graph is *proper* if any of its edges spans just two consecutive levels. Several papers about constructing level drawings of level graphs assume that the input graph is proper. Otherwise, they suggest to make it proper by "simply adding dummy vertices" along the edges spanning more than two levels. In this paper we show that this apparently innocent augmentation has dramatic consequences if, instead of constructing just a level drawing, we are also interested in representing additional constraints, like clustering of vertices or consecutivity constraints on the ordering of vertices on levels.

A *level graph* $G = (V, E, \gamma)$ is a graph with a function $\gamma : V \rightarrow \{1, 2, ..., k\}$, with $1 \leq k \leq |V|$ such that $\gamma(u) \neq \gamma(v)$ for each edge $(u, v) \in E$. The set $V_i = \{v|\gamma(v) = i\}$ is the i-th *level* of G. A level graph $G = (V, E, \gamma)$ is *proper* if for every edge $(u, v) \in E$, it holds $\gamma(u) = \gamma(v) \pm 1$. A *level planar drawing* of (V, E, γ) maps each vertex v of each level V_i to a point on line $y = i$, denoted by L_i, and each edge to a y-monotone curve between its endpoints so that no two edges intersect. A level graph is *level planar* if it admits a level planar drawing. A linear-time algorithm for testing level planarity was presented by Jünger and Leipert in [10].

A *clustered-level graph* (*cl-graph*) (V, E, γ, T) is a level graph (V, E, γ) equipped with a *cluster hierarchy* T, that is, a rooted tree where each leaf is an element of V and each internal node μ, called *cluster*, represents the subset V_μ of V composed of the leaves of the subtree of T rooted at μ. A *clustered-level planar drawing* (*cl-planar* drawing) of (V, E, γ, T) is a level planar drawing of level graph (V, E, γ) such that: (1) each cluster μ is represented by a simple region enclosing all and only the vertices in V_μ; (2) no edge intersects the boundary of a cluster more than once; (3) no two cluster boundaries intersect each other; and (4) the intersection of L_i with any cluster μ is a straight-line segment, that is, the vertices of V_i that belong to μ are consecutive along L_i. A cl-graph is *clustered-level planar* (*cl-planar*) if it admits a cl-planar drawing. CLUSTERED-LEVEL PLANARITY (CL-PLANARITY) is the problem of testing whether a given cl-graph is cl-planar. The CL-PLANARITY problem was introduced by Forster and Bachmaier [9], who showed a polynomial-time testing algorithm for the case in which the level graph is a proper hierarchy and the clusters are level-connected.

A $\mathcal{T}$*-level graph* (also known as *generalized k-ary tanglegram*) $(V, E, \gamma, \mathcal{T})$ is a level graph (V, E, γ) equipped with a set $\mathcal{T} = T_1, \ldots, T_k$ of trees such that the leaves of T_i are the vertices of level V_i of (V, E, γ), for $1 \leq i \leq k$. A $\mathcal{T}$*-level planar drawing* of $(V, E, \gamma, \mathcal{T})$ is a level planar drawing of (V, E, γ) such that, for $i = 1, \ldots, k$, the order in which the vertices of V_i appear along L_i is *compatible* with T_i, that is, for each node w of T_i, the leaves of the subtree of T_i rooted at w appear consecutively along L_i. A $\mathcal{T}$-level graph is $\mathcal{T}$*-level planar* if it admits a $\mathcal{T}$-level planar drawing. T-LEVEL PLANARITY is the problem of testing whether a given $\mathcal{T}$-level graph is $\mathcal{T}$-level planar. The

* Research was supported in part by the Italian Ministry of Education, University, and Research (MIUR) under PRIN 2012C4E3KT national research project "AMANDA – Algorithmics for MAssive and Networked DAta" and by ESF project 10-EuroGIGA-OP-003 GraDR. Fabrizio Frati was partially supported by the Australian Research Council (grant DE140100708)'.

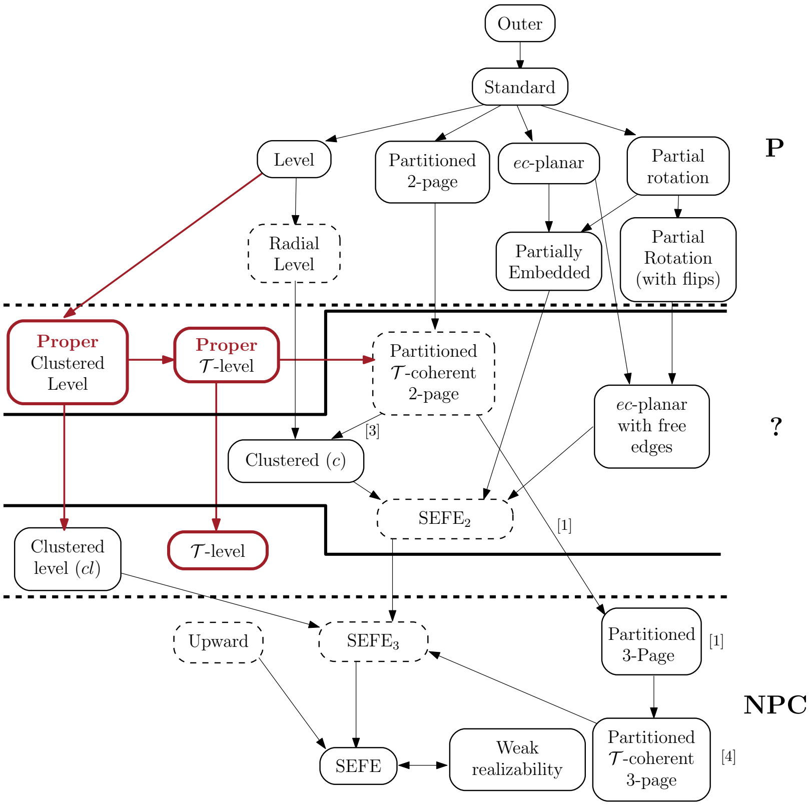
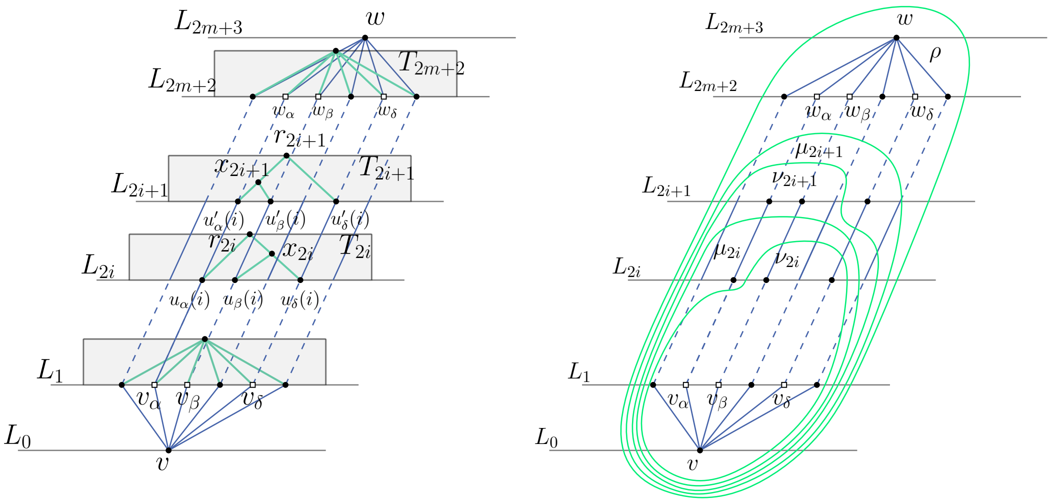


Fig. 1. Updates on the classification proposed by Schaefer in [12]. Dashed lines represent the boundaries between problems that were known to be polynomial-time solvable, problems that were known to be $\mathcal{NP}$-complete, and problems whose complexity was unknown before this paper. Solid lines represent the new boundaries according to the results of this paper. All the arcs representing reductions that can be transitively inferred are omitted. Results proved after [12] are equipped with references. Reductions and classes introduced in this paper are red. The prefix "proper" has been added to two classes in [12] to better clarify their nature.

T-LEVEL PLANARITY problem was introduced by Wotzlaw *et al.* [13], who showed a quadratic-time algorithm for the case in which the number of vertices at each level is bounded by a constant.

The definition of *proper* naturally extends to cl-graphs and $\mathcal{T}$-level graphs. Note that, given any non-proper level graph G it is easy to construct a proper level graph G' that is level planar if and only if G is level planar. However, as mentioned above, there exists no trivial transformation from a non-proper cl-graph (a non-proper $\mathcal{T}$-level graph) to an equivalent proper cl-graph (resp., an equivalent proper $\mathcal{T}$-level graph).

In this paper we show that CLUSTERED-LEVEL PLANARITY and T-LEVEL PLANARITY are $\mathcal{NP}$-complete for non-proper instances. Conversely, we show that both problems are polynomial-time solvable for proper instances. Our results have several consequences: (1) They narrow the gap between polynomiality and $\mathcal{NP}$-completeness in the classification of Schaefer [12] (see Fig. 1). The reduction of Schaefer between T-LEVEL PLANARITY and SEFE-2 holds for proper instances [12]. (2) They allow to partially answer a question from [12] asking whether a reduction exists from CL-PLANARITY to SEFE-2. We show that such a reduction exists for proper instances and that a reduction from general instances would imply the $\mathcal{NP}$-hardness of SEFE-2. (3) They improve on [9] and [13] by extending the classes of instances which are decidable in polynomial-time for CL-PLANARITY and T-LEVEL PLANARITY, respectively. (4) They provide the first, as far as we know, $\mathcal{NP}$-completeness for a problem that has all the constraints of clustered planarity problem (and some more).



(a)

(b)

Fig. 2. Illustrations for the proof of (a) Theorem 1 and (b) Theorem 2.

The paper is organized as follows. The $\mathcal{NP}$-completeness proofs are in Section 2, while the algorithms are in Section 3. We conclude with open problems in Section 4.

2 NP-Hardness

In this section we prove that the T-LEVEL PLANARITY and the CL-PLANARITY problems are $\mathcal{NP}$-complete. In both cases, the $\mathcal{NP}$-hardness is proved by means of a polynomial-time reduction from the $\mathcal{NP}$-complete problem BETWEENNESS [11], that takes as input a finite set A of n objects and a set C of m ordered triples of distinct elements of A, and asks whether a linear ordering $\mathcal{O}$ of the elements of A exists such that for each triple $\langle \alpha, \beta, \delta \rangle$ of C, we have either $\mathcal{O} =<\dots, \alpha, \dots, \beta, \dots, \delta, \dots>$ or $\mathcal{O} =<\dots, \delta, \dots, \beta, \dots, \alpha, \dots>$.

Theorem 1. *T-LEVEL PLANARITY is $\mathcal{NP}$-complete.*

Proof: The problem trivially belongs to $\mathcal{NP}$. We prove the $\mathcal{NP}$-hardness. Given an instance $\langle A, C \rangle$ of BETWEENNESS, we construct an equivalent instance $(V, E, \gamma, \mathcal{T})$ of T-LEVEL PLANARITY as follows. Let $A = \{1, 2, \dots, n\}$ and let $m = |C|$. Graph (V, E) is composed of a set of paths connecting two vertices v and w. Refer to Fig. 2(a).

Initialize $V = \{v, w\}$ and $E = \emptyset$, with $\gamma(v) = 0$ and $\gamma(w) = 2m + 3$. Let $T_0 \in \mathcal{T}$ and $T_{2m+3} \in \mathcal{T}$ be trees with a single node v and w, respectively.

For each $j = 1, \dots, n$, add two vertices v_j and w_j to V, with $\gamma(v_j) = 1$ and $\gamma(w_j) = 2m + 2$. Add edges (v, v_j) and (w, w_j) to E. Also, let $T_1 \in \mathcal{T}$ and $T_{2m+2} \in \mathcal{T}$ be two stars whose leaves are all the vertices of levels V_1 and V_{2m+2}, respectively. Further, for each $j = 1, \dots, n$, we initialize variable $last(j) = v_j$.

Then, for each $i = 1, \dots, m$, consider the triple $t_i = \langle \alpha, \beta, \gamma \rangle$. Add six vertices $u_\alpha(i)$, $u'_\alpha(i)$, $u_\beta(i)$, $u'_\beta(i)$, $u_\delta(i)$, and $u'_\delta(i)$ to V with $\gamma(u_\alpha(i)) = \gamma(u_\beta(i)) = \gamma(u_\delta(i)) = 2i$ and $\gamma(u'_\alpha(i)) = \gamma(u'_\beta(i)) = \gamma(u'_\delta(i)) = 2i + 1$. Also, add edges $(last(\alpha), u_\alpha(i))$, $(last(\beta), u_\beta(i))$, $(last(\delta), u_\delta(i))$, $(u_\alpha(i), u'_\alpha(i))$, $(u_\beta(i), u'_\beta(i))$, and $(u_\gamma(i), u'_\gamma(i))$ to E. Further, set $last(\alpha) = u'_\alpha(i)$, $last(\beta) = u'_\beta(i)$, and $last(\delta) = u'_\delta(i)$. Let $T_{2i} \in \mathcal{T}$ be a binary tree with a root r_{2i}, an internal node x_{2i} and a leaf $u_\alpha(i)$ both adjacent to r_{2i}, and with leaves $u_\beta(i)$ and $u_\delta(i)$ both adjacent to x_{2i}. Moreover, let $T_{2i+1} \in \mathcal{T}$ be a binary tree with a root r_{2i+1}, an internal node x_{2i+1} and a leaf $u'_\delta(i)$ both adjacent to r_{2i+1}, and with leaves $u'_\alpha(i)$ and $u'_\beta(i)$ both adjacent to x_{2i+1}.

Finally, for each $j = 1, \dots, n$, add an edge $(last(j), w_j)$ to E.

The reduction is easily performed in $O(n + m)$ time. We prove that $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar if and only if $\langle A, C \rangle$ is a positive instance of BETWEENNESS.

Suppose that $(V, E, \gamma, \mathcal{T})$ admits a $\mathcal{T}$-level planar drawing Γ. Consider the left-to-right order $\mathcal{O}_1$ in which the vertices of level V_1 appear along L_1. Construct an order $\mathcal{O}$ of the elements of A such that $\alpha \in A$ appears before $\beta \in A$ if and only if $v_\alpha \in V_1$ appears before $v_\beta \in V_1$ in $\mathcal{O}_1$. In order to prove that $\mathcal{O}$ is a positive solution for $\langle A, C\rangle$, it suffices to prove that, for each triple $t_i = \langle \alpha, \beta, \delta\rangle \in C$, vertices v_α, v_β, and v_δ appear either in this order or in the reverse order in $\mathcal{O}_1$. Note that tree T_{2i} enforces $u_\alpha(i)$ not to lie between $u_\beta(i)$ and $u_\delta(i)$ along L_{2i}; also, tree T_{2i+1} enforces $u'_\delta(i)$ not to lie between $u'_\alpha(i)$ and $u'_\beta(i)$ along L_{2i+1}. Since the three paths connecting v and w and passing through v_α, v_β, and v_δ do not cross each other in Γ and since they contain $u_\alpha(i)$ and $u'_\alpha(i)$, $u_\beta(i)$ and $u'_\beta(i)$, and $u_\delta(i)$ and $u'_\delta(i)$, respectively, we have that v_α, v_β, and v_δ appear either in this order or in the reverse order in $\mathcal{O}_1$.

Suppose that an ordering $\mathcal{O}$ of the elements of A exists that is a positive solution of BETWEENNESS for instance $\langle A, C\rangle$. In order to construct Γ, place the vertices of V_1 and V_{2m+2} along L_1 and L_{2m+2} in such a way that vertices $v_j \in V_1$ and $w_j \in V_{2m+2}$, for $j = 1, \dots, n$, are assigned x-coordinate equal to s if j is the s-th element of $\mathcal{O}$. Also, for $i = 1, \dots, m$, let $t_i = \langle \alpha, \beta, \delta\rangle \in C$. Place vertices $u_\lambda(i)$ and $u'_\lambda(i)$, with $\lambda \in \{\alpha, \beta, \delta\}$, on L_{2i} and L_{2i+1}, respectively, in such a way that $u_\lambda(i)$ and $u'_\lambda(i)$ are assigned x-coordinate equal to s if λ is the s-th element of $\mathcal{O}$. Finally, place v and w at any points on L_0 and L_{2m+3}, respectively, and draw the edges of E as straight-line segments. We prove that Γ is a $\mathcal{T}$-level planar drawing of $(V, E, \gamma, \mathcal{T})$. First note that, by construction, Γ is a level planar drawing of (V, E, γ). Further, for each $i = 1, \dots, m$, vertices $u_\alpha(i)$, $u_\beta(i)$, and $u_\delta(i)$ appear along L_{2i} either in this order or in the reverse order; in both cases, the order is compatible with tree T_{2i}. Analogously, vertices $u'_\alpha(i)$, $u'_\beta(i)$, and $u'_\delta(i)$ appear along L_{2i+1} either in this order or in the reverse order; in both cases, the order is compatible with tree T_{2i+1}. Finally, the order in which vertices of V_0, V_1, V_{2m+2}, and V_{2m+3} appear along L_0, L_1, L_{2m+2}, and L_{2m+3}, respectively, are trivially compatible with T_0, T_1, T_{2m+2}, and T_{2m+3}. □

Note that the reduction described in Theorem 1 can be modified in such a way that $\mathcal{T}$ contains only binary trees by removing levels V_1 and V_{2m+2}. Indeed, the presence of these two levels was only meant to simplify the description of the relationship between the order of the elements of A and the order of the paths between v and w.

Theorem 2. CLUSTERED-LEVEL PLANARITY *is $\mathcal{NP}$-complete.*

Proof: The problem trivially belongs to class $\mathcal{NP}$. We prove the $\mathcal{NP}$-hardness. Given an instance $\langle A, C\rangle$ of BETWEENNESS, we construct an instance $(V, E, \gamma, \mathcal{T})$ of T-LEVEL PLANARITY as in the proof of Theorem 1; then, starting from $(V, E, \gamma, \mathcal{T})$, we construct an instance (V, E, γ, T) of CL-PLANARITY that is cl-planar if and only if $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar. This, together with the fact that $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar if and only if $\langle A, C\rangle$ is a positive instance of BETWEENNESS, implies the $\mathcal{NP}$-hardness of CL-PLANARITY. Refer to Fig. 2(b).

Cluster hierarchy T is constructed as follows. Initialize T with a root ρ. Let $w \in V_{2m+3}$ and $w_j \in V_{2m+2}$, for $j = 1, \dots, n$, be leaves of T that are children of ρ; add an internal node μ_{2m+1} to T as a child of ρ. Next, for $i = m, \dots, 1$, let $u'_\delta(i)$ be a leaf of T that is child of μ_{2i+1}; add an internal node ν_{2i+1} to T as a child of μ_{2i+1}; then, let $u'_\alpha(i)$ and $u'_\beta(i)$ be leaves of T that are children of ν_{2i+1}; add an internal node μ_{2i} to T as a child of ν_{2i+1}. Further, let $u_\alpha(i)$ be a leaf of T that is a child of μ_{2i}; add an internal node ν_{2i} to T as a child of μ_{2i}; then, let $u_\beta(i)$ and $u_\delta(i)$ be leaves of T that are children of ν_{2i}; add an internal node μ_{2i-1} to T as a child of ν_{2i}. Finally, let vertices $v \in V_0$ and $v_j \in V_1$, for $j = 1, \dots, n$, be leaves of T that are children of μ_1.

We prove that (V, E, γ, T) is cl-planar if and only if $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar.

Suppose that (V, E, γ, T) admits a cl-planar drawing Γ. Construct a $\mathcal{T}$-level planar drawing Γ^* of $(V, E, \gamma, \mathcal{T})$ by removing from Γ the clusters of T. First, observe that the drawing of (V, E, γ) in Γ^* is level-planar, since it is level-planar in Γ. Further, for each $i = 1, \dots, m$, vertex $u_\alpha(i)$ does not appear between $u_\beta(i)$ and $u_\gamma(i)$ along line L_{2i}, since $u_\beta(i), u_\gamma(i) \in \nu_{2i}$ and $u_\alpha(i) \notin \nu_{2i}$; analogously, vertex $u'_\delta(i)$ does not appear between $u'_\alpha(i)$ and $u'_\beta(i)$ along line L_{2i+1}, since $u'_\alpha(i), u'_\beta(i) \in \nu_{2i+1}$ and $u'_\delta(i) \notin \nu_{2i+1}$. Hence, the order of the vertices of V_{2i} and V_{2i+1} along L_{2i} and L_{2i+1}, respectively, are compatible with trees T_{2i} and T_{2i+1}. Finally, the order in which vertices of V_0, V_1, V_{2m+2}, and V_{2m+3} appear along lines L_0, L_1, L_{2m+2}, and L_{2m+3}, respectively, are trivially compatible with T_0, T_1, T_{2m+2}, and T_{2m+3}.

Suppose that $(V, E, \gamma, \mathcal{T})$ admits a $\mathcal{T}$-level planar drawing Γ^*; we describe how to construct a cl-planar drawing Γ of (V, E, γ, T). Assume that Γ^* is a straight-line drawing, which is not a loss of generality [8]. Initialize $\Gamma = \Gamma^*$. Draw each cluster α in T as a convex region $R(\alpha)$ in Γ slightly surrounding the border of the convex hull of its vertices and slightly surrounding the border of the regions representing the clusters that are its descendants in T. Let j be the largest index such that V_j contains a vertex of α. Then, $R(\alpha)$ contains all and only the vertices that are descendants of

α in T; moreover, any two clusters α and β in T are one contained into the other, hence $R(\alpha)$ and $R(\beta)$ do not cross; finally, we prove that no edge e in E crosses more than once the boundary of $R(\alpha)$ in Γ. First, if at least one end-vertex of e belongs to α, then e and the boundary of $R(\alpha)$ cross at most once, given that e is a straight-line segment and that $R(\alpha)$ is convex. All the vertices in $V_0 \cup \ldots \cup V_{j-1}$ and at least two vertices of V_j belong to α, hence their incident edges do not cross the boundary of $R(\alpha)$ more than once. Further, all the vertices in $V_{j+1} \cup \ldots \cup V_{2m+3}$ have y-coordinates larger than every point of $R(\alpha)$, hence edges between them do not cross $R(\alpha)$. It remains to consider the case in which e connects a vertex x_1 in V_j not in α (there is at most one such vertex) with a vertex x_2 in $V_{j+1} \cup \ldots \cup V_{2m+2}$; in this case e and $R(\alpha)$ do not cross given that x_1 is outside $R(\alpha)$, that x_2 has y-coordinate larger than every point of $R(\alpha)$, and that $R(\alpha)$ is arbitrarily close to the convex hull of its vertices. □

3 Polynomial-Time Algorithms

In this section we prove that problems T-LEVEL PLANARITY and CL-PLANARITY become polyomial-time solvable if restricted to proper instances.

3.1 T-LEVEL PLANARITY

We start by describing a polynomial-time algorithm for T-LEVEL PLANARITY. The algorithm is based on a reduction to the *Simultanoues Embedding with Fixed Edges* problem for two graphs (SEFE-2), that is defined as follows.

A *simultanoues embedding with fixed edges* (SEFE) of two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ on the same set of vertices V consists of two planar drawings Γ_1 and Γ_2 of G_1 and G_2, respectively, such that each vertex $v \in V$ is mapped to the same point in both drawings and each edge of the *common graph* $G_\cap = (V, E_1 \cap E_2)$ is represented by the same simple curve in the two drawings. The SEFE-2 problem asks whether a given pair of graphs $\langle G_1, G_2 \rangle$ admits a SEFE [5]. The computational complexity of the SEFE-2 problem is unknown, but there exist polynomial-time algorithms for instances that respect some conditions [2,5,6,7,12]. We are going to use a result by Bläsius and Rütter [7], who proposed a quadratic-time algorithm for instances $\langle G_1, G_2 \rangle$ of SEFE-2 in which G_1 and G_2 are 2-connected, and the common graph $G_\cap$ is connected.

In the analysis of the complexity of the following algorithms we assume that the internal nodes of the trees in $\mathcal{T}$ in any instance $(V, E, \gamma, \mathcal{T})$ of T-LEVEL PLANARITY and of tree T in any instance (V, E, γ, T) of CL-PLANARITY have at least two children. It is easily proved that this is not a loss of generality; also, this allows us to describe the size of the instances in terms of the size of their sets of vertices.

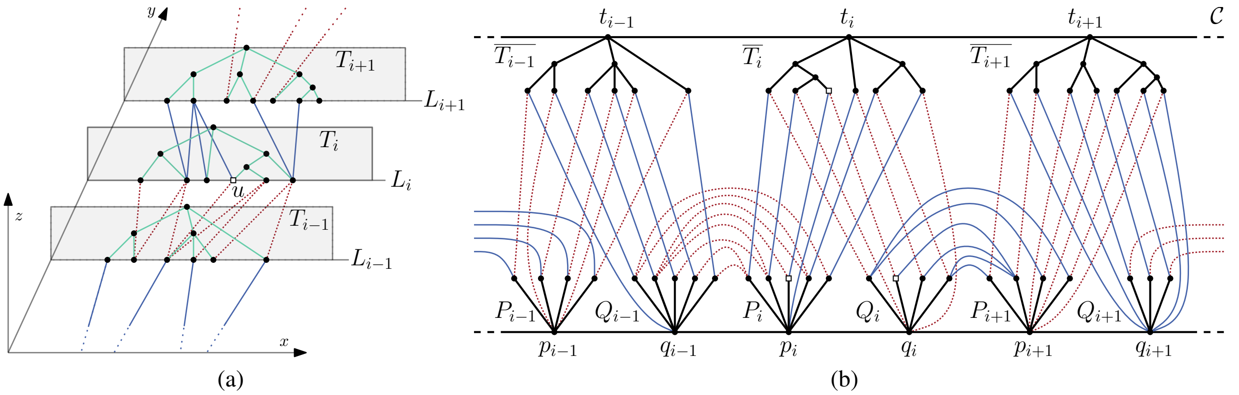
Lemma 1. *Let $(V, E, \gamma, \mathcal{T})$ be a proper instance of T-LEVEL PLANARITY. There exists an equivalent instance $\langle G_1^*, G_2^* \rangle$ of SEFE-2 such that $G_1^* = (V^*, E_1^*)$ and $G_2^* = (V^*, E_2^*)$ are 2-connected, and the common graph $G_\cap = (V^*, E_1^* \cap E_2^*)$ is connected. Further, instance $\langle G_1^*, G_2^* \rangle$ can be constructed in linear time.*

Proof: We describe how to construct instance $\langle G_1^*, G_2^* \rangle$. Refer to Fig. 3.

Graph $G_\cap$ contains a cycle $\mathcal{C} = t_1, t_2, \ldots, t_k, q_k, p_k, q_{k-1}, p_{k-1}, \ldots, q_1, p_1$, where k is the number of levels of $(V, E, \gamma, \mathcal{T})$. For each $i = 1, \ldots, k$, graph $G_\cap$ contains a copy $\overline{T_i}$ of tree $T_i \in \mathcal{T}$, whose root is identified with vertex t_i, and contains two stars P_i and Q_i centered at vertices p_i and q_i, respectively, whose number of leaves is as follows. For each vertex $u \in V_i$ such that an edge $(u, v) \in E$ exists connecting u to a vertex $v \in V_{i-1}$, star P_i contains a leaf vertex $u(P_i)$; also, for each vertex $u \in V_i$ such that an edge $(u, v) \in E$ exists connecting u to a vertex $v \in V_{i+1}$, star Q_i contains a leaf vertex $u(Q_i)$. We also denote by $u(\overline{T_i})$ a leaf of $\overline{T_i}$ corresponding to vertex $u \in V_i$.

Graph G_1^* contains $G_\cap$ plus a set of edges defined as follows. For $i = 1, \ldots, k$, consider each vertex $u \in V_i$. Suppose that i is even. Then, G_1^* has an edge connecting the leaf $u(\overline{T_i})$ of $\overline{T_i}$ corresponding to u with either the leaf $u(Q_i)$ of Q_i corresponding to u, if it exists, or with the center q_i of Q_i, otherwise; also, for each edge in E connecting a vertex $u \in V_i$ with a vertex $v \in V_{i-1}$, graph G_1^* has an edge connecting the leaf $u(P_i)$ of P_i corresponding to u with the leaf $v(Q_{i-1})$ of Q_{i-1} corresponding to v (such leaves exist by construction). Suppose that i is odd. Then, graph G_1^* has an edge between $u(\overline{T_i})$ and either $u(P_i)$, if it exists, or the center p_i of P_i, otherwise.

Graph G_2^* contains $G_\cap$ plus a set of edges defined as follows. For $i = 1, \ldots, k$, consider each vertex $u \in V_i$. Suppose that i is odd. Then, G_2^* has an edge connecting $u(\overline{T_i})$ with either the leaf $u(Q_i)$ of Q_i corresponding to u,



(a) (b)

Fig. 3. Illustration for the proof of Lemma 1. Index i is assumed to be even. (a) A $\mathcal{T}$-level planar drawing Γ of instance $(V, E, \gamma, \mathcal{T})$. (b) The SEFE $\langle \Gamma_1, \Gamma_2 \rangle$ of instance $\langle G_1^*, G_2^* \rangle$ of SEFE-2 corresponding to Γ. Correspondence between a vertex $u \in V_i$ and leaves $u(\overline{T_i}) \in \overline{T_i}$, $u(P_i) \in P_i$, and $u(Q_i) \in Q_i$ is highlighted by representing all such vertices as white boxes.

if it exists, or with the center q_i of Q_i, otherwise; also, for each edge in E connecting a vertex $u \in V_i$ with a vertex $v \in V_{i-1}$, graph G_2^* has an edge $(u(P_i), v(Q_{i-1}))$. Suppose that i is even. Then, graph G_2^* has an edge between $u(\overline{T_i})$ and either $u(P_i)$, if it exists, or p_i, otherwise.

It is easy to see that $G_\cap$ is connected and that $\langle G_1^*, G_2^* \rangle$ can be constructed in polynomial time. We prove that G_1^* and G_2^* are 2-connected, that is, removing any vertex v disconnects neither G_1^* nor G_2^*. If v is a leaf of either $\overline{T_i}$ or P_i or Q_i, with $1 \le i \le k$, then removing v disconnects neither G_1^* nor G_2^*, since $G_\cap$ remains connected. If v is an internal node (the root) of $\overline{T_i}$, or P_i, or Q_i, say of $\overline{T_i}$, with $1 \le i \le k$, then removing v disconnects $G_\cap$ into $m = \deg(v)$ (resp. $m = \deg(v) - 1$) components, namely one component $\overline{T_i}(v)$ containing all the vertices of $\mathcal{C}$ (resp. all the vertices of $\mathcal{C}$, except for v) and $m - 1$ subtrees $\overline{T_i}^j$ of $\overline{T_i}$, with $j = 1, \dots, m-1$, rooted the children of v; however, by construction, each $\overline{T_i}^j$ is connected to $\overline{T_i}(v)$ via at least an edge $(u(\overline{T_i}), u(P_i)) \in E_1^*$ and an edge $(u(\overline{T_i}), u(Q_i)) \in E_2^*$, or vice versa, incident to one of its leaves $u(\overline{T_i})$.

Observe that, if $(V, E, \gamma, \mathcal{T})$ has $n_\mathcal{T}$ nodes in the trees of $\mathcal{T}$ (where $|V| < n_\mathcal{T}$), then $\langle G_1^*, G_2^* \rangle$ contains at most $3n_\mathcal{T}$ vertices. Also, the number of edges of $\langle G_1^*, G_2^* \rangle$ is at most $|E| + 2n_\mathcal{T}$. Hence, the size of $\langle G_1^*, G_2^* \rangle$ is linear in the size of $(V, E, \gamma, \mathcal{T})$ and it is easy to see that $\langle G_1^*, G_2^* \rangle$ can be constructed in linear time.

We prove that $\langle G_1^*, G_2^* \rangle$ admits a SEFE if and only if $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar.

Suppose that $\langle G_1^*, G_2^* \rangle$ admits a SEFE $\langle \Gamma_1^*, \Gamma_2^* \rangle$. We show how to construct a drawing Γ of $(V, E, \gamma, \mathcal{T})$. For $1 \le i \le k$, let $\Theta(\overline{T_i})$ be the order in which the leaves of $\overline{T_i}$ appear in a pre-order traversal of $\overline{T_i}$ in $\langle \Gamma_1^*, \Gamma_2^* \rangle$; then, let the ordering $\mathcal{O}_i$ of the vertices of V_i along L_i be either $\Theta(\overline{T_i})$, if i is odd, or the reverse of $\Theta(\overline{T_i})$, if i is even.

We prove that Γ is $\mathcal{T}$-level planar. For each $i = 1, \dots, k$, $\mathcal{O}_i$ is compatible with $T_i \in \mathcal{T}$, since the drawing of $\overline{T_i}$, that belongs to $G_\cap$, is planar in $\langle \Gamma_1^*, \Gamma_2^* \rangle$. Suppose, for a contradiction, that two edges $(u, v), (w, z) \in E$ exist, with $u, w \in V_i$ and $v, z \in V_{i+1}$, that intersect in Γ. Hence, either u appears before w in $\mathcal{O}_i$ and v appears after z in $\mathcal{O}_{i+1}$, or vice versa. Since i and $i+1$ have different parity, either u appears before w in $\Theta(\overline{T_i})$ and v appears before z in $\Theta(\overline{T_{i+1}})$, or vice versa. We claim that, in both cases, this implies a crossing in $\langle \Gamma_1^*, \Gamma_2^* \rangle$ between paths $(q_i, u(Q_i), v(P_{i+1}), p_{i+1})$ and $(q_i, w(Q_i), z(P_{i+1}), p_{i+1})$ in $\langle G_1^*, G_2^* \rangle$. Since the edges of these two paths belong all to G_1^* or all to G_2^*, depending on whether i is even or odd, this yields a contradiction. We now prove the claim. The pre-order traversal $\Theta(Q_i)$ of Q_i (the pre-order traversal $\Theta(P_{i+1})$ of P_{i+1}) in $\langle \Gamma_1^*, \Gamma_2^* \rangle$ restricted to the leaves of Q_i (of P_{i+1}) is the reverse of $\Theta(\overline{T_i})$ (of $\Theta(\overline{T_{i+1}})$) restricted to the vertices of V_i (of V_{i+1}) corresponding to leaves of Q_i (of P_{i+1}). Namely, each leaf $x(Q_i)$ of Q_i ($y(P_{i+1})$ of P_{i+1}) is connected to leaf $x(\overline{T_i})$ of $\overline{T_i}$ ($y(\overline{T_{i+1}})$ of $\overline{T_{i+1}}$) in the same graph, either G_1^* or G_2^*, by construction. Hence, the fact that u appears before (after) w in $\Theta(\overline{T_i})$ and v appears before (after) z in $\Theta(\overline{T_{i+1}})$ implies that u appears after (before) w in $\Theta(Q_i)$ and v appears after (before) z in $\Theta(P_{i+1})$. In both cases, this implies a crossing in $\langle \Gamma_1^*, \Gamma_2^* \rangle$ between the two paths.

Suppose that $(V, E, \gamma, \mathcal{T})$ admits a $\mathcal{T}$-level planar drawing Γ. We show how to construct a SEFE $\langle \Gamma_1^*, \Gamma_2^* \rangle$ of $\langle G_1^*, G_2^* \rangle$. For $1 \leq i \leq k$, consider the order $\mathcal{O}_i$ of the vertices of level V_i along L_i in Γ. Since Γ is $\mathcal{T}$-level planar, there exists an embedding Γ_i of tree $T_i \in \mathcal{T}$ that is compatible with $\mathcal{O}_i$. If i is odd (even), then assign to each internal vertex of $\overline{T_i}$ the same (resp. the opposite) rotation scheme as its corresponding vertex in Γ_i. Also, if i is odd, then assign to p_i (to q_i) the rotation scheme in G_1^* (in G_2^*) such that the paths connecting p_i (q_i) to the leaves of $\overline{T_i}$ (either with an edge or passing through a leaf of the corresponding star of $G_\cap$) appear in the same clockwise order as the vertices of V_i appear in $\mathcal{O}_i$; if i is even, then assign to p_i (to q_i) the rotation scheme in G_2^* (in G_1^*) such that the paths connecting p_i (q_i) to the leaves of $\overline{T_i}$ appear in the same counterclockwise order as the vertices of V_i appear in $\mathcal{O}_i$. Finally, consider the embedding $\Gamma_{i,i+1}$ obtained by restricting Γ to the vertices and edges of the subgraph induced by the vertices of V_i and V_{i+1}. If i is odd (even), then assign to the leaves of Q_i and of P_{i+1} in G_1^* (in G_2^*) the same rotation scheme as their corresponding vertices have in $\Gamma_{i,i+1}$. This completes the construction of $\langle \Gamma_1^*, \Gamma_2^* \rangle$.

We prove that $\langle \Gamma_1^*, \Gamma_2^* \rangle$ is a SEFE of $\langle G_1^*, G_2^* \rangle$. Since the rotation scheme of the internal vertices of each $\overline{T_i}$ are constructed starting from an embedding of Γ_i of tree $T_i \in \mathcal{T}$ that is compatible with $\mathcal{O}_i$, the drawing of $\overline{T_i}$ is planar. Further, since the rotation schemes of p_i (of q_i) are also constructed starting from $\mathcal{O}_i$, there exists no crossing between two paths connecting t_i and p_i (t_i and q_i), one passing through a leaf $u(\overline{T_i})$ of $\overline{T_i}$ and, possibly, through a leaf $u(P_i)$ of P_i (through a leaf $u(Q_i)$ of Q_i), and the other passing through a leaf $v(\overline{T_i})$ of $\overline{T_i}$ and, possibly, through a leaf $v(P_i)$ of P_i (through a leaf $v(Q_i)$ of Q_i). Finally, since the rotation schemes of the leaves of Q_i and P_{i+1} are constructed from the embedding $\Gamma_{i,i+1}$ obtained by restricting Γ to the vertices and edges of the subgraph induced by the vertices of V_i and V_{i+1}, there exist no two crossing edges between leaves of Q_i and of P_{i+1}. □

We remark that a reduction from T-LEVEL PLANARITY to SEFE-2 was described by Schaefer in [12]; however, the instances of SEFE-2 obtained from that reduction do not satisfy any conditions that make SEFE-2 known to be solvable in polynomial-time.

Theorem 3. *Let $(V, E, \gamma, \mathcal{T})$ be a proper instance of* T-LEVEL PLANARITY. *There exists a quadratic-time algorithm that decides whether $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar.*

Proof: By Lemma 1, an instance $\langle G_1, G_2 \rangle$ of SEFE-2 can be constructed in linear time such that G_1 and G_2 are 2-connected, the common graph $G_\cap$ is connected, and $\langle G_1, G_2 \rangle$ is a positive instance of SEFE-2 if and only if $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar. The statement follows from the fact that there exists a quadratic-time algorithm [7] that decides whether $\langle G_1, G_2 \rangle$ is a positive instance of SEFE-2. □

3.2 CLUSTERED-LEVEL PLANARITY

In the following we prove that the polynomial-time algorithm to decide the existence of a $\mathcal{T}$-level planar drawing of a proper instance $(V, E, \gamma, \mathcal{T})$ of T-LEVEL PLANARITY can be also employed to decide in polynomial time the existence of a cl-planar drawing of a proper instance (V, E, γ, T) of CL-PLANARITY.

A proper cl-graph (V, E, γ, T) is *μ-connected between two levels* V_i and V_{i+1} if there exist two vertices $u \in V_\mu \cap V_i$ and $v \in V_\mu \cap V_{i+1}$ such that edge $(u, v) \in E$. For a cluster $\mu \in T$, let $\gamma_{\min}(\mu) = \min\{i | V_i \cap V_\mu \neq \emptyset\}$ and let $\gamma_{\max}(\mu) = \max\{i | V_i \cap V_\mu \neq \emptyset\}$. A proper cl-graph (V, E, γ, T) is *level-μ-connected* if it is μ-connected between levels V_i and V_{i+1} for each $i = \gamma_{\min}(\mu), \dots, \gamma_{\max}(\mu) - 1$. A proper cl-graph (V, E, γ, T) is *level-connected* if it is μ-level-connected for each cluster $\mu \in T$.

Our strategy consists of first transforming a proper instance of CL-PLANARITY into an equivalent level-connected instance, and then transforming such a level-connected instance into an equivalent proper instance of T-LEVEL PLANARITY.

Lemma 2. *Let (V, E, γ, T) be a proper instance of* CLUSTERED-LEVEL PLANARITY. *There exists an equivalent level-connected instance $(V^*, E^*, \gamma^*, T^*)$ of* CLUSTERED-LEVEL PLANARITY. *Further, the size of $(V^*, E^*, \gamma^*, T^*)$ is quadratic in the size of (V, E, γ, T) and $(V^*, E^*, \gamma^*, T^*)$ can be constructed in quadratic time.*

Proof: The construction of $(V^*, E^*, \gamma^*, T^*)$ works in two steps. See Fig. 4.

First, we transform (V, E, γ, T) into an equivalent instance (V', E', γ', T'). Initialize $V' = V$, $E' = E$, and $T' = T$. Also, for each $i = 1, \dots, k$ and for each vertex $u \in V_i$, set $\gamma'(u) = 3(i-1)+1$. Then, for each $i = 1, \dots, k-1$,

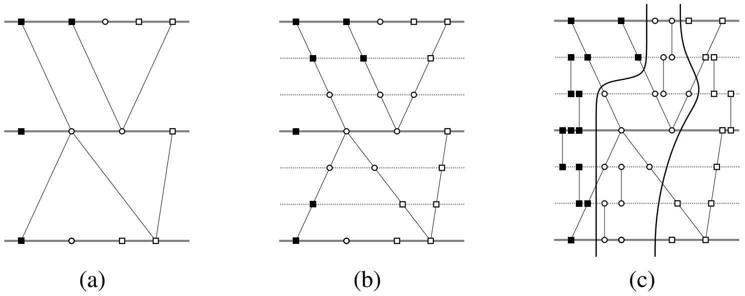


Fig. 4. Illustration for the proof of Lemma 2. (a) Instance (V, E, γ, T) with flat hierarchy containing clusters $\mu_{\blacksquare}$, $\mu_{\square}$, and $\mu_{\circ}$. (b) Insertion of dummy vertices in (V, E, γ, T) to obtain (V', E', γ', T'). (c) Level-connected instance $(V^*, E^*, \gamma^*, T^*)$ obtained from (V', E', γ', T').

consider each edge $(u, v) \in E$ such that $\gamma(u) = i$ and $\gamma(v) = i + 1$. Add two vertices d_u and d_v to V', and replace (u, v) in E' with three edges (u, d_u), (d_u, d_v), and (d_v, v). Set $\gamma'(d_u) = 3(i-1)+2$ and $\gamma'(d_v) = 3i$. Finally, add d_u (d_v) to T' as a child of the parent of u (of v) in T'.

Second, we transform (V', E', γ', T') into an equivalent level-connected instance $(V^*, E^*, \gamma^*, T^*)$. Initialize $V^* = V'$, $E^* = E'$, $\gamma^* = \gamma'$, and $T^* = T'$. Consider each cluster $\mu \in T'$ according to a bottom-up visit of T'. If there exists a level V'_i, with $\gamma'_{\min}(\mu) \leq i < \gamma'_{\max}(\mu)$, such that no edge in E' connects a vertex $u \in V'_i \cap V'_\mu$ with a vertex $v \in V'_{i+1} \cap V'_\mu$, then add two vertices u^* and v^* to V^*, add an edge (u^*, v^*) to E^*, set $\gamma^*(u^*) = i$ and $\gamma^*(v^*) = i + 1$, and add u^* and v^* to T^* as children of μ.

Observe that, for each cluster $\mu \in T$ and for each level $1 \leq i \leq 3k - 2$, at most two dummy vertices are added to $(V^*, E^*, \gamma^*, T^*)$. This implies that $|V^*| \in O(|V|^2)$. Also, the whole construction can be performed in $O(|V|^2)$ time.

Claim 1 *(V', E', γ', T') is equivalent to (V, E, γ, T).*

Proof: Suppose that (V, E, γ, T) admits a cl-planar drawing Γ; we show how to construct a cl-planar drawing Γ' of (V', E', γ', T'). Initialize $\Gamma' = \Gamma$. We scale Γ' up by a factor of 3 and we vertically translate it so that the vertices in V'_1 lie on line $y = 1$. After the two affine transformations have been applied (i) no crossing has been introduced in the drawing, (ii) every edge is still drawn as a y-monotone curve, (iii) for $i = 1, \dots, k$, the vertices of level $V_i = V'_{3(i-1)+1}$ are placed on line $y = 3(i-1)+1$, that we denote by $L'_{3(i-1)+1}$, and (iv) the order in which vertices of $V_i = V'_{3(i-1)+1}$ appear along $L'_{3(i-1)+1}$ is the same as the order in which they appeared along L_i. For each $i = 1, \dots, k-1$, consider each edge $(u, v) \in E$ such that $\gamma(u) = i$ and $\gamma(v) = i + 1$. Place vertices d_u and d_v in Γ' on the two points of the curve representing (u, v) having y-coordinate $3(i-1)+2$ and $3i$, respectively. Then, the curves representing in Γ' any two edges in E' are part of the curves representing in Γ' any two edges in E. Hence Γ' is a cl-planar drawing of (V', E', γ', T').

Suppose that (V', E', γ', T') admits a cl-planar drawing Γ'; we show how to construct a cl-planar drawing Γ of (V, E, γ, T). Initialize $\Gamma = \Gamma'$. For each $i = 1, \dots, k-1$, consider each path (u, d_u, d_v, v) such that $\gamma'(u) = 3(i-1)+1$, $\gamma'(d_u) = 3(i-1)+2$, $\gamma'(d_v) = 3i$, and $\gamma'(v) = 3i+1$; remove vertices d_u and d_v, and their incident edges in E' from Γ; draw edge $(u, v) \in E$ as a curve obtained as a composition of the curves representing edges (u, d_u), (d_u, d_v), and (d_v, v) in Γ'. Scale Γ down by a factor of 3 and vertically translate it so that the vertices of V_1 lie on line $y = 1$. After the two affine transformations have been applied (i) no crossing has been introduced in the drawing, (ii) every edge is still drawn as a y-monotone curve, (iii) for $i = 1, \dots, k$, the vertices of level V_i are placed on line $y = i$, and (iv) the order in which vertices of $V_i = V'_{3(i-1)+1}$ appear along L_i is the same as the order in which they appeared along $L'_{3(i-1)+1}$. Since Γ' is cl-planar, this implies that Γ is cl-planar, as well. □

Instance (V', E', γ', T') is such that, if there exists a vertex $u \in V'_j$, with $1 \leq j \leq 3(k-1)+1$, that is adjacent to two vertices $v, w \in V'_h$, with $h = j \pm 1$, then u, v, and w have the same parent node $\mu \in T'$; hence, (V', E', γ', T') is μ-connected between levels V'_j and V'_h.

Claim 2 *$(V^*, E^*, \gamma^*, T^*)$ is equivalent to (V', E', γ', T').*

Proof: Suppose that $(V^*, E^*, \gamma^*, T^*)$ admits a cl-planar drawing Γ^*; we show how to construct a cl-planar drawing Γ' of (V', E', γ', T'). Initialize $\Gamma' = \Gamma^*$ and remove from V', E', and Γ' all the vertices and edges added when constructing Γ^*. Since all the other vertices of V' and edges of E' have the same representation in Γ' and in Γ^*, and since Γ^* is cl-planar, drawing Γ' is cl-planar, as well.

Suppose that (V', E', γ', T') admits a cl-planar drawing Γ'; we show how to construct a cl-planar drawing Γ^* of $(V^*, E^*, \gamma^*, T^*)$. Initialize $\Gamma^* = \Gamma'$. Consider a level V'_i, with $1 \leq i \leq 3(k-1)$, such that vertices $u^*, v^* \in \mu$ with $\gamma'(u^*) = i$ and $\gamma'(v^*) = i+1$, for some cluster $\mu \in T$, have been added to $(V^*, E^*, \gamma^*, T^*)$. By construction, (V', E', γ', T') is not μ-connected between levels V'_i and V'_{i+1}. As observed before, this implies that no vertex $u \in V'_i \cap V'_\mu$ exists that is connected to two vertices $v, w \in V'_{i+1}$, and no vertex $u \in V'_{i+1} \cap V'_\mu$ exists that is connected to two vertices $v, w \in V'_i$. Hence, vertices u^* and v^*, and edge (u^*, v^*), can be drawn in Γ^* entirely inside the region representing μ in such a way that u^* and v^* lie along lines L'_i and L'_{i+1} and there exists no crossing between edge (u^*, v^*) and another edge. □

This concludes the proof of the lemma. □

Lemma 3. *Let (V, E, γ, T) be a level-connected instance of* Clustered-Level Planarity. *There exists an equivalent proper instance $(V, E, \gamma, \mathcal{T})$ of* T-Level Planarity. *Further, the size of $(V, E, \gamma, \mathcal{T})$ is linear in the size of (V, E, γ, T) and $(V, E, \gamma, \mathcal{T})$ can be constructed in quadratic time.*

Proof: We construct $(V, E, \gamma, \mathcal{T})$ from (V, E, γ, T) as follows. Initialize $\mathcal{T} = \emptyset$. For $i = 1, \ldots, k$, add to $\mathcal{T}$ a tree T_i that is the subtree of the cluster hierarchy T whose leaves are all and only the vertices of level V_i. Note that the set of leaves of the trees in $\mathcal{T}$ corresponds to the vertex set V. Since each internal node of the trees in $\mathcal{T}$ has at least two children, we have that the size of $(V, E, \gamma, \mathcal{T})$ is linear in the size of (V, E, γ, T). Also, the construction of $(V, E, \gamma, \mathcal{T})$ can be easily performed in $O(|V|^2)$ time.

We prove that $(V, E, \gamma, \mathcal{T})$ is $\mathcal{T}$-level planar if and only if (V, E, γ, T) is cl-planar.

Suppose that $(V, E, \gamma, \mathcal{T})$ admits a $\mathcal{T}$-level planar drawing Γ^*; we show how to construct a cl-planar drawing Γ of (V, E, γ, T). Initialize $\Gamma = \Gamma^*$. Consider each level V_i, with $i = 1, \ldots, k$. By construction, for each cluster $\mu \in T$ such that there exists a vertex $v \in V_i \cap V_\mu$, there exists an internal node of tree $T_i \in \mathcal{T}$ whose leaves are all and only the vertices of $V_i \cap V_\mu$. Since Γ^* is $\mathcal{T}$-level planar, such vertices appear consecutively along L_i. Hence, in order to prove that Γ is a cl-planar drawing, it suffices to prove that there exist no four vertices u, v, w, z such that (i) $u, v \in V_i$ and $w, z \in V_j$, with $1 \leq i < j \leq k$; (ii) $u, w \in V_\mu$ and $v, z \in V_\nu$, with $\mu \neq \nu$; and (iii) u appears before v on L_i and w appears after z on L_j, or vice versa. Suppose, for a contradiction, that such four vertices exist. Note that, we can assume $j = i \pm 1$ without loss of generality, as (V, E, γ, T) is level-connected. Assume that u appears before v along L_i and w appears after z along L_j, the other case being symmetric. Since Γ^* is $\mathcal{T}$-level planar, all the vertices of V_μ appear before all the vertices of V_ν along L_i and all the vertices of V_μ appear after all the vertices of V_ν along L_j. Also, since (V, E, γ, T) is level-connected, there exists at least an edge (a, b) such that $a \in V_i \cap V_\mu$ and $b \in V_j \cap V_\mu$, and an edge (c, d) such that $c \in V_i \cap V_\nu$ and $d \in V_j \cap V_\nu$. However, under the above conditions, these two edges intersect in Γ and in Γ^*, hence contradicting the hypothesis that Γ^* is $\mathcal{T}$-level planar.

Suppose that (V, E, γ, T) admits a cl-planar drawing Γ; we show how to construct a $\mathcal{T}$-level planar drawing Γ^* of $(V, E, \gamma, \mathcal{T})$. Initialize $\Gamma^* = \Gamma$. Consider each level V_i, with $i = 1, \ldots, k$. By construction, for each internal node w of tree $T_i \in \mathcal{T}$, there exists a cluster $\mu \in T$ such that the vertices of $V_i \cap V_\mu$ are all and only the leaves of the subtree of T_i rooted at w. Since Γ is cl-planar, such vertices appear consecutively along L_i. Hence, Γ^* is $\mathcal{T}$-level planar. □

We get the following.

Theorem 4. *Let (V, E, γ, T) be a proper instance of* Clustered-Level Planarity. *There exists an $O(|V|^4)$-time algorithm that decides whether (V, E, γ, T) admits a cl-planar drawing.*

Proof: By Lemma 2, it is possible to construct in $O(|V|^2)$ time a level-connected instance (V', E', γ', T') of CL-Planarity that is cl-planar if and only if (V, E, γ, T) is cl-planar, with $|V'| = O(|V|^2)$. By Lemma 3, it is possible to construct in $O(|V'|^2)$ time a proper instance $(V', E', \gamma', \mathcal{T}')$ of T-Level Planarity that is $\mathcal{T}$-level planar if and only if (V', E', γ', T') is cl-planar. Finally, by Theorem 3, it is possible to test in $O(|V'|^2)$ time whether $(V', E', \gamma', \mathcal{T}')$ is $\mathcal{T}$-level planar. □

4 Open Problems

Several problems are opened by this research:

1. The algorithm in [10] for testing level planarity and the algorithm in [9] for testing CL-PLANARITY for level-connected instances in which the level graph is a proper hierarchy both have linear-time complexity. The algorithm in [13] for testing T-LEVEL PLANARITY for instances in which the number of vertices on each level is bounded by a constant has quadratic-time complexity. Although our polynomial-time algorithms solve more general problems than the ones cited above, they are less efficient. Hence, there is room for future research aiming at improving our complexity bounds.
2. Our $\mathcal{NP}$-completeness result on the complexity of CL-PLANARITY exploits a cluster hierarchy whose depth is linear in the number of vertices of the underlying graph. Does the $\mathcal{NP}$-hardness hold even when the hierarchy is flat or has a depth that is sublinear in the number of vertices?
3. The $\mathcal{NP}$-hardness of CL-PLANARITY presented in this paper is, to the best of our knowledge, the first hardness result for a variation of the clustered planarity problem in which none of the c-planarity constraints is dropped. Is it possible to use similar techniques to tackle the more intriguing problem of determining the complexity of CLUSTERED PLANARITY?

References

1. Angelini, P., Da Lozzo, G., Neuwirth, D.: On the complexity of some problems related to SEFE. CoRR abs/1207.3934 (2013)
2. Angelini, P., Di Battista, G., Frati, F., Patrignani, M., Rutter, I.: Testing the simultaneous embeddability of two graphs whose intersection is a biconnected or a connected graph. J. of Discrete Algorithms 14, 150–172 (2012)
3. Angelini, P., Da Lozzo, G.: Deepening the relationship between SEFE and C-planarity. CoRR abs/1404.6175 (2014)
4. Angelini, P., Da Lozzo, G., Neuwirth, D.: On some NP-complete SEFE problems. In: Pal, S.P., Sadakane, K. (eds.) WALCOM. LNCS, vol. 8344, pp. 200–212 (2014)
5. Blasiüs, T., Kobourov, S.G., Rutter, I.: Simultaneous embedding of planar graphs. In: Tamassia, R. (ed.) Handbook of Graph Drawing and Visualization. CRC Press (2013)
6. Bläsius, T., Rutter, I.: Disconnectivity and relative positions in simultaneous embeddings. In: Didimo, W., Patrignani, M. (eds.) GD'13. LNCS, vol. 7704, pp. 31–42 (2013)
7. Bläsius, T., Rutter, I.: Simultaneous PQ-ordering with applications to constrained embedding problems. In: Khanna, S. (ed.) SODA. pp. 1030–1043. SIAM (2013)
8. Eades, P., Feng, Q.W., Lin, X., Nagamochi, H.: Straight-line drawing algorithms for hierarchical graphs and clustered graphs. Algorithmica 44(1), 1–32 (2006)
9. Forster, M., Bachmaier, C.: Clustered level planarity. In: van Emde Boas, P., Pokorný, J., Bieliková, M., Stuller, J. (eds.) SOFSEM. LNCS, vol. 2932, pp. 218–228 (2004)
10. Jünger, M., Leipert, S.: Level planar embedding in linear time. J. Graph Algorithms Appl. 6(1), 67–113 (2002)
11. Opatrny, J.: Total ordering problem. SIAM J. Comput. 8(1), 111–114 (1979)
12. Schaefer, M.: Toward a theory of planarity: Hanani-Tutte and planarity variants. J. of Graph Alg. and Appl 17(4), 367–440 (2013)
13. Wotzlaw, A., Speckenmeyer, E., Porschen, S.: Generalized k-ary tanglegrams on level graphs: A satisfiability-based approach and its evaluation. Discrete Applied Mathematics 160(16-17), 2349–2363 (2012)

Semi-algebraic Ramsey numbers

Andrew Suk[*]

Abstract

Given a finite point set $P \subset \mathbb{R}^d$, a k-ary semi-algebraic relation E on P is the set of k-tuples of points in P, which is determined by a finite number of polynomial equations and inequalities in kd real variables. The description complexity of such a relation is at most t if the number of polynomials and their degrees are all bounded by t. The Ramsey number $R_k^{d,t}(s,n)$ is the minimum N such that any N-element point set P in $\mathbb{R}^d$ equipped with a k-ary semi-algebraic relation E, such that E has complexity at most t, contains s members such that every k-tuple induced by them is in E, or n members such that every k-tuple induced by them is not in E.

We give a new upper bound for $R_k^{d,t}(s,n)$ for $k \geq 3$ and s fixed. In particular, we show that for fixed integers d, t, s

$$R_3^{d,t}(s,n) \leq 2^{n^{o(1)}},$$

establishing a subexponential upper bound on $R_3^{d,t}(s,n)$. This improves the previous bound of 2^{n^C} due to Conlon, Fox, Pach, Sudakov, and Suk, where C is a very large constant depending on d, t, and s. As an application, we give new estimates for a recently studied Ramsey-type problem on hyperplane arrangements in $\mathbb{R}^d$. We also study multi-color Ramsey numbers for triangles in our semi-algebraic setting, achieving some partial results.

1 Introduction

Classical Ramsey numbers. A k-uniform hypergraph $H = (P, E)$ consists of a vertex set P and an edge set $E \subset \binom{P}{k}$, which is a collection of subsets of P of size k. The Ramsey number $R_k(s,n)$ is the minimum integer N such that every k-uniform hypergraph on N vertices contains either s vertices such that every k-tuple induced by them is an edge, or contains n vertices such that every k-tuple induced by them is not an edge.

Due to a wide range of applications in logic, number theory, analysis, and geometry, estimating Ramsey numbers has become one of the most central problems in combinatorics. For *diagonal* Ramsey numbers, i.e. when $s = n$, the best known lower and upper bounds on $R_k(n,n)$ are of the form[1] $R_2(n,n) = 2^{\Theta(n)}$, and for $k \geq 3$,

$$\mathrm{twr}_{k-1}(\Omega(n^2)) \leq R_k(n,n) \leq \mathrm{twr}_k(O(n)),$$

where the tower function $\mathrm{twr}_k(x)$ is defined by $\mathrm{twr}_1(x) = x$ and $\mathrm{twr}_{i+1} = 2^{\mathrm{twr}_i(x)}$ (see [20, 17, 19, 18]). Erdős, Hajnal, and Rado [19] conjectured that $R_k(n,n) = \mathrm{twr}_k(\Theta(n))$, and Erdős offered a

[*] Massachusetts Institute of Technology, Cambridge, MA. Supported by an NSF Postdoctoral Fellowship and by Swiss National Science Foundation Grant 200021-125287/1. Email: asuk@math.mit.edu.

[1] We write $f(n) = O(g(n))$ if $|f(n)| \leq c|g(n)|$ for some fixed constant c and for all $n \geq 1$; $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$; and $f(n) = \Theta(g(n))$ if both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ hold. We write $f(n) = o(g(n))$ if for every positive $\epsilon > 0$ there exists a constant n_0 such that $f(n) \leq \epsilon|g(n)|$ for all $n \geq n_0$.

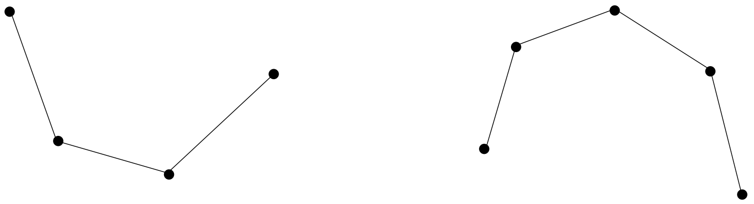


Figure 1: A 4-cup and a 5-cap.

\$500 reward for a proof. Despite much attention over the last 50 years, the exponential gap between the lower and upper bounds on $R_k(n,n)$, for $k \geq 3$, remains unchanged.

The *off-diagonal* Ramsey numbers, i.e. $R_k(s,n)$ with s fixed and n tending to infinity, has also been extensively studied. Unlike $R_k(n,n)$, the lower and upper bounds on $R_k(s,n)$ are much more comparable. It is known [4, 25, 7, 8] that $R_2(3,n) = \Theta(n^2/\log n)$ and, for fixed $s > 3$

$$\Omega\left(n^{\frac{s+1}{2}-\epsilon}\right) \leq R_2(s,n) \leq O\left(n^{s-1}\right), \tag{1}$$

where $\epsilon > 0$ is an arbitrarily small constant. Combining the upper bound in (1) with the results of Erdős, Hajnal, and Rado [18, 19] demonstrates that

$$\mathrm{twr}_{k-1}(\Omega(n)) \leq R_k(s,n) \leq \mathrm{twr}_{k-1}(O(n^{2s-4})), \tag{2}$$

for $k \geq 3$ and $s \geq 2^k$. See Conlon, Fox, and Sudakov [14] for a recent improvement.

Semi-algebraic setting. In this paper, we continue a sequence of recent works on Ramsey numbers for k-ary semi-algebraic relations E on $\mathbb{R}^d$ (see [9, 16, 13, 33]). Before we give its precise definition, let us recall two classic Ramsey-type theorems of Erdős and Szekeres.

Theorem 1.1 ([20]). *For $N = (s-1)(n-1)+1$, let $P = (p_1, ..., p_N) \subset \mathbb{R}$ be a sequence of N distinct real numbers. Then P contains either an increasing subsequence of length s, or a decreasing subsequence on length n.*

In fact, there are now at least 6 different proofs of Theorem 1.1 (see [32]). The other well-known result from [20] is the following theorem, which is often referred to as the Erdős-Szekeres cups-caps Theorem. Let X be a finite point set in the plane in general position.[2] We say that $X = (p_{i_1}, ..., p_{i_s})$ forms an *s-cup* (*s-cap*) if X is in convex position and its convex hull is bounded above (below) by a single edge. See Figure 1.

Theorem 1.2 ([20]). *For $N = \binom{n+s-4}{s-2} + 1$, let $P = (p_1, ..., p_N)$ be a sequence of N points in the plane in general position. Then P contains either an s-cup or an n-cap.*

Theorems 1.1 and 1.2 can be generalized using following semi-algebraic framework. Let $P = \{p_1, ..., p_N\}$ be a sequence of N points in $\mathbb{R}^d$. Then we say that $E \subset \binom{P}{k}$ is a *semi-algebraic relation* on P with *complexity* at most t, if there are t polynomials $f_1, ..., f_t \in \mathbb{R}[x_1, ..., x_{kd}]$ of degree at most t, and a Boolean function Φ such that for $1 \leq i_1 < \cdots < i_k \leq N$,

$$(p_{i_1}, ..., p_{i_k}) \in E \quad \Leftrightarrow \quad \Phi(f_1(p_{i_1}, ..., p_{i_k}) \geq 0, ..., f_t(p_{i_1}, ..., p_{i_k}) \geq 0) = 1.$$

[2] No two members share the same x-coordinate, and no three members are collinear.

We say that the relation $E \subset \binom{P}{k}$ is *symmetric* if $(p_{i_1}, ..., p_{i_k}) \in E$ iff for all permutation π,

$$\Phi(f_1(p_{\pi(i_1)}, ..., p_{\pi(i_k)}) \geq 0, ..., f_t(p_{\pi(i_1)}, ..., p_{\pi(i_k)}) \geq 0) = 1.$$

Point sets $P \subset \mathbb{R}^d$ equipped with a k-ary semi-algebraic relation $E \subset \binom{P}{k}$ are often used to model problems in discrete geometry, where the dimension d, uniformity k, and complexity t are considered fixed but arbitrarily large constants. Since we can always make any relation E symmetric by increasing its complexity to $t' = t'(k, d, t)$, we can therefore simplify our presentation by only considering symmetric relations.

Let $R_k^{d,t}(s, n)$ be the minimum integer N such that every N-element point set P in $\mathbb{R}^d$ equipped with a k-ary (symmetric) semi-algebraic relation $E \subset \binom{P}{k}$, which has complexity at most t, contains s points such that every k-tuple induced by them is in E, or contains n points such that no k-tuple induced by them is in E. Alon, Pach, Pinchasi, Radoičić, and Sharir [5] showed that for $k = 2$, we have

$$R_2^{d,t}(n, n) \leq n^C, \tag{3}$$

where $C = C(d, t)$. Roughly speaking, $C \approx t\binom{d+t}{t}$. Conlon, Fox, Pach, Sudakov, and Suk showed that one can adapt the Erdős-Rado argument in [18] and establish the following recursive formula for $R_k^{d,t}(s, n)$.

Theorem 1.3 ([13])**.** *Set $M = R_{k-1}^{d,t}(s-1, n-1)$. Then for every $k \geq 3$,*

$$R_k^{d,t}(s, n) \leq 2^{C_1 M \log M},$$

where $C_1 = C_1(k, d, t)$.

Together with (3) we have $R_k^{d,t}(n, n) \leq \mathrm{twr}_{k-1}(n^C)$, giving an exponential improvement over the Ramsey numbers for general k-uniform hypergraphs. Conlon et al. [13] also gave a construction of a geometric example that provides a $\mathrm{twr}_{k-1}(\Omega(n))$ lower bound, demonstrating that $R_k^{d,t}(n, n)$ does indeed grow as a $(k-1)$-fold exponential tower in n.

However, off-diagonal Ramsey numbers for semi-algebraic relations are much less understood. The best known upper bound for $R_k^{d,t}(s, n)$ is essential the trivial bound

$$R_k^{d,t}(s, n) \leq \min\left\{R_k^{d,t}(n, n), R_k(s, n)\right\}.$$

The crucial case is when $k = 3$, since any significant improvement on estimating $R_3^{d,t}(s, n)$ could be used with Theorem 1.3 to obtain a better bound for $R_k^{d,t}(s, n)$, for $k \geq 4$. The trivial bound implies that $R_3^{d,t}(s, n) \leq 2^{n^C}$, where C is a large constant depending on d, t, and s.

Our main result establishes the following improved upper bound for $R_3^{d,t}(s, n)$, showing that the function $R_3^{d,t}(s, n)$ is indeed subexponential in n.

Theorem 1.4. *For fixed integers $d, t \geq 1$ and $s \geq 4$, we have $R_3^{d,t}(s, n) \leq 2^{n^{o(1)}}$.*

Combining Theorems 1.4 and 1.3 we have the following.

Corollary 1.5. *For fixed integers $d, t \geq 1$, $k \geq 3$, and $s \geq k+1$, we have*

$$R_k^{d,t}(s, n) \leq \mathrm{twr}_{k-1}(n^{o(1)}).$$

For $d \geq 2$ and $t \geq 1$, the classic cups-caps construction of Erdős and Szekeres [20] shows that $R_3^{d,t}(s,n) \geq \Omega(n^{s-2})$, and together with the semi-algebraic stepping-up lemma proven in [13] (see also [27]) we have $R_k^{d,t}(s,n) \geq \mathrm{twr}_{k-2}(\Omega(n^{s-2}))$ for $s, d \geq 2^k$.

In Section 5, we give an application of Theorem 1.4 to a recently studied problem on hyperplane arrangements in $\mathbb{R}^d$.

Monochromatic triangles. Let $R_2(\underbrace{s,...,s}_{m})$ denote the smallest integer N such that any m-coloring on the edges of the complete N-vertex graph contains a monochromatic clique of size s, that is, a set of s vertices such that every pair from this set has the same color. For the case $s = 3$, the Ramsey number $R_2(\underbrace{3,...,3}_{m})$ has received a lot of attention over the last 100 years due to its application in additive number theory [31] (more details are given in Section 6.1). It is known (see [23, 31]) that

$$\Omega(3.19^m) \leq R_2(\underbrace{3,...,3}_{m}) \leq O(m!).$$

Our next result states that we can improve the upper bound on $R_2(3,...,3)$ in our semi-algebraic setting. More precisely, let $R_2^{d,t}(\underbrace{3,...,3}_{m})$ be the minimum integer N such that every N-element point set P in $\mathbb{R}^d$ equipped with symmetric semi-algebraic relations $E_1,...,E_m \subset \binom{P}{2}$, such that each E_i has complexity at most t and $\binom{P}{2} = E_1 \cup \cdots \cup E_m$, contains three points such that every pair induced by them belongs to E_i for some fixed i.

Theorem 1.6. *For fixed $d, t \geq 1$ we have*

$$R_2^{d,t}(\underbrace{3,...,3}_{m}) < 2^{O(m \log\log m)}.$$

We also show that for fixed $d \geq 1$ and $t \geq 5000$, the function $R_2^{d,t}(\underbrace{3,...,3}_{m})$ does indeed grow exponential in m.

Theorem 1.7. *For $d \geq 1$ and $t \geq 5000$ we have*

$$R_2^{d,t}(\underbrace{3,...,3}_{m}) \geq c(1681)^{m/7} \geq c(2.889)^m,$$

where c is an absolute constant.

Organization. In the next two sections, we recall several old theorems on the arrangement of surfaces in $\mathbb{R}^d$ and establish a result on point sets equipped with multiple binary relations. In Section 4, we combine the results from Sections 2 and 3 to prove our main result, Theorem 1.4. We discuss a short proof of our application in Section 5, and our results on monochromatic triangles in Section 6. We conclude with some remarks.

We systemically omit floor and ceiling signs whenever they are not crucial for the sake of clarity of our presentation. All logarithms are assumed to be base 2.

2 Arrangement of surfaces in $\mathbb{R}^d$

In this section, we recall several old results on the arrangement of surfaces in $\mathbb{R}^d$. Let $f_1, ..., f_m$ be d-variate real polynomials of degree at most t, with zero sets $Z_1, ..., Z_m$, that is, $Z_i = \{x \in \mathbb{R}^d : f_i(x) = 0\}$. Set $\Sigma = \{Z_1, ..., Z_m\}$. We will assume that d and t are fixed, and m is some number tending to infinity. A *cell* in the arrangement $\mathcal{A}(\Sigma) = \bigcup_i Z_i$ is a relatively open connected set defined as follows. Let $\approx$ be an equivalence relation on $\mathbb{R}^d$, where $x \approx y$ if $\{i : x \in Z_i\} = \{i : y \in Z_i\}$. Then the cells of the arrangement Σ are the connected components of the equivalence classes. A vector $\sigma \in \{-1, 0, +1\}^m$ is a *sign pattern* of $f_1, ..., f_m$ if there exists an $x \in \mathbb{R}^d$ such that the sign of $f_j(x)$ is σ_j for all $j = 1, ..., m$. The Milnor-Thom theorem (see [6, 30, 34]) bounds the number of cells in the arrangement of the zero sets $Z_1, ..., Z_m$ and, consequently, the number of possible sign patterns.

Theorem 2.1 (Milnor-Thom)**.** *Let $f_1, ..., f_m$ be d-variate real polynomials of degree at most t. The number of cells in the arrangement of their zero sets $Z_1, ..., Z_m \subset \mathbb{R}^d$ and, consequently, the number of sign patterns of $f_1, ..., f_m$ is at most*

$$\left(\frac{50mt}{d}\right)^d,$$

for $m \geq d \geq 1$.

While the Milnor-Thom Theorem bounds the number of cells in the arrangement $\mathcal{A}(\Sigma)$, the complexity of these cells may be very large (depending on m). A long standing open problem is whether each cell can be further decomposed into semi-algebraic sets[3] with bounded description complexity (which depends only on d and t), such that the total number of cells for the whole arrangement is still $O(m^d)$. This can be done easily in dimension 2 by a result of Chazelle et al. [11]. Unfortunately in higher dimensions, the current bounds for this problem are not tight. In dimension 3, Chazelle et al. [11] established a near tight bound of $O(m^3\beta(m))$, where $\beta(m)$ is an extremal slowly growing function of m related to the inverse Ackermann function. For dimensions $d \geq 4$, Koltun [26] established a general bound of $O(m^{2d-4+\epsilon})$ for arbitrarily small constant ϵ, which is nearly tight in dimension 4. By combining these bounds with the standard theory of random sampling [3, 12, 5], one can obtain the following result which is often referred to as the Cutting Lemma. We say that the surface $Z_i = \{x \in \mathbb{R}^d : f_i(x) = 0\}$ *crosses* the cell $\Delta \subset \mathbb{R}^d$ if $Z_i \cap \Delta \neq \emptyset$ and Z_i does not fully contain Δ.

Lemma 2.2 (Cutting Lemma)**.** *For $d, t \geq 1$, let Σ be a family of m algebraic surfaces (zero sets) in $\mathbb{R}^d$ of degree at most t. Then for any $r > 0$, there exists a decomposition of $\mathbb{R}^d$ into at most $c_1 r^{2d}$ relatively open connected sets (cells), where $c_1 = c_1(d, t)$, such that each cell is crossed by at most m/r surfaces from Σ.*

As an application, we prove the following lemma (see [28, 10] for a similar result when Σ is a collection of hyperplanes).

Lemma 2.3. *For $d, t \geq 1$, let P be an N-element point set in $\mathbb{R}^d$ and let Σ be a family of m surfaces of degree at most t. Then for any integer $\ell > \log m$, we can find ℓ disjoint subsets P_i and ℓ cells Δ_i, with $\Delta_i \supset P_i$, such that each subset P_i contains at least $N/(4\ell)$ points from P, and every surface in Σ crosses at most $c_2\ell^{1-1/(2d)}$ cells Δ_i, where $c_2 = c_2(d, t)$.*

[3] A real semi-algebraic set in $\mathbb{R}^d$ is the locus of all points that satisfy a given finite Boolean combination of polynomial equations and inequalities in the d coordinates.

Proof. We first find Δ_1 and P_1 as follows. Let $\ell > \log m$ and let c_1 be as defined in Lemma 2.2. Given a family Σ of m surfaces in $\mathbb{R}^d$, we apply Lemma 2.2 with parameter $r = (\ell/c_1)^{1/(2d)}$, and decompose $\mathbb{R}^d$ into at most ℓ cells, such that each cell is crossed by at most $\frac{m}{(\ell/c_1)^{1/(2d)}}$ surfaces from Σ. By the pigeonhole principle, there is a cell Δ_1 that contains least N/ℓ points from P. Let P_1 be a subset of exactly $\lfloor N/\ell \rfloor$ points in $\Delta_1 \cap P$. Now for each surface from Σ that crosses Δ_1, we "double it" by adding another copy of that surface to our collection. This gives us a new family of surfaces Σ_1 such that

$$|\Sigma_1| \leq m + \frac{m}{(\ell/c_1)^{1/(2d)}} = m\left(1 + \frac{1}{(\ell/c_1)^{1/(2d)}}\right).$$

After obtaining subsets $P_1, ..., P_i$ such that $|P_j| = \lfloor \frac{N}{\ell}(1 - \frac{1}{\ell})^{j-1} \rfloor$ for $1 \leq j \leq i$, cells $\Delta_1, ..., \Delta_i$, and the family of surfaces Σ_i such that

$$|\Sigma_i| \leq m\left(1 + \frac{1}{(\ell/c_1)^{1/(2d)}}\right)^i,$$

we obtain P_{i+1}, Δ_{i+1}, Σ_{i+1} as follows. Given Σ_i, we apply Lemma 2.2 with the same parameter $r = (\ell/c_1)^{1/(2d)}$, and decompose $\mathbb{R}^d$ into at most ℓ cells, such that each cell is crossed by at most $\frac{|\Sigma_i|}{(\ell/c_1)^{1/(2d)}}$ surfaces from Σ_i. Let $P' = P \setminus (P_1 \cup \cdots \cup P_i)$. By the pigeonhole principle, there is a cell Δ_{i+1} that contains at least

$$\begin{aligned}
\tfrac{|P'|}{\ell} &\geq \left(N - \sum_{j=1}^{i} \tfrac{N}{\ell}(1 - \tfrac{1}{\ell})^{j-1}\right)/\ell \\
&= \tfrac{N}{\ell}\left(1 - \tfrac{1}{\ell}\sum_{j=1}^{i}(1 - \tfrac{1}{\ell})^{j-1}\right) \\
&= \tfrac{N}{\ell}\left(1 - \tfrac{1}{\ell}\right)\left(1 - \tfrac{1}{\ell} - \tfrac{1}{\ell}\sum_{j=1}^{i-1}(1 - \tfrac{1}{\ell})^{j-1}\right) \\
&= \tfrac{N}{\ell}\left(1 - \tfrac{1}{\ell}\right)^i
\end{aligned}$$

points from P'. Let P_{i+1} be a subset of exactly $\lfloor \frac{N}{\ell}(1 - 1/\ell)^i \rfloor$ points in $\Delta_{i+1} \cap P'$. Finally, for each surface from Σ_i that crosses Δ_{i+1}, we "double it" by adding another copy of that surface to our collection, giving us a new family of surfaces Σ_{i+1} such that

$$\begin{aligned}
|\Sigma_{i+1}| &\leq |\Sigma_i| + \tfrac{|\Sigma_i|}{(\ell/c_1)^{1/(2d)}} \\
&= |\Sigma_i|\left(1 + \tfrac{1}{(\ell/c_1)^{1/(2d)}}\right) \\
&= m\left(1 + \tfrac{1}{(\ell/c_1)^{1/(2d)}}\right)^{i+1}.
\end{aligned}$$

Notice that $|P_i| \geq N/(4\ell)$ for $i \leq \ell$. Once we have obtained subsets $P_1, ..., P_\ell$ and cell $\Delta_1, ..., \Delta_\ell$, it is easy to see that each surface in Σ crosses at most $O(r^{1-1/(2d)})$ cells Δ_i. Indeed suppose $Z \in \Sigma$ crosses κ cells. Then by the arguments above, there must be 2^κ copies of Z in Σ_ℓ. Hence we have

$$2^{\kappa} \leq m\left(1+\frac{1}{(\ell/c_1)^{1/(2d)}}\right)^{\ell} \leq me^{c_1\ell^{1-1/(2d)}}.$$

Since $\ell \geq \log m$, we have

$$\kappa \leq c_2\ell^{1-1/(2d)},$$

for sufficiently large $c_2 = c_2(d,t)$.

□

3 Multiple binary relations

Let P be a set of N points in $\mathbb{R}^d$, and let $E_1,...,E_m \subset \binom{P}{2}$ be semi-algebraic relations on P such that E_i has complexity at most t. The goal of this section is to find a large subset $P' \subset P$ such that $\binom{P'}{2} \cap E_i = \emptyset$ for all i, given that the clique number in the graphs $G_i = (P,E_i)$ are small.

First we recall classic theorem of Dilworth (see also [22]). Let $G = (V,E)$ be a graph whose vertices are ordered $V = \{v_1,...,v_N\}$. We say that E is *transitive* on V if for $1 \leq i_1 < i_2 < i_3 \leq N$, $(v_{i_1},v_{i_2}),(v_{i_2},v_{i_3}) \in E$ implies that $(v_{i_1},v_{i_3}) \in E$.

Theorem 3.1 (Dilworth). *Let $G = (V,E)$ be an N-vertex graph whose vertices are ordered $V = \{v_1,...,v_N\}$, such that E is transitive on V. If G has clique number ω, then G contains an independent set of size N/ω.*

Lemma 3.2. *For integers $m \geq 2$ and $d,t \geq 1$, let P be a set of N points in $\mathbb{R}^d$ equipped with (symmetric) semi-algebraic relations $E_1,...,E_m \subset \binom{P}{2}$, where each E_i has complexity at most t. Then there is a subset $P' \subset P$ of size $N^{1/(c_3\log m)}$, where $c_3 = c_3(d,t)$, and a fixed ordering on P' such that each relation E_i is transitive on P'.*

Proof. We proceed by induction on N. Let c_3 be a sufficiently large number, depending only on d and t, that will be determined later. For each relation $E_i \subset \binom{P}{2}$, let $f_{i,1},...,f_{i,t}$ be polynomials of degree at most t and let Φ_i be a boolean function such that

$$(p,q) \in E_i \quad \Leftrightarrow \quad \Phi_i(f_{i,1}(p,q) \geq 0,...,f_{i,t}(p,q) \geq 0) = 1.$$

For each $p \in P$, $i \in \{1,...,m\}$, and $j \in \{1,...,t\}$, we define the surface $Z_{p,i,j} = \{x \in \mathbb{R}^d : f_{i,j}(p,x) = 0\}$. Then let Σ be the family of Nmt surfaces in $\mathbb{R}^d$ defined by

$$\Sigma = \{Z_{p,i,j} : p \in P, 1 \leq i \leq m, 1 \leq j \leq t\}.$$

By applying Lemma 2.2 to Σ with parameter $r = (mt)^2$, there is a decomposition of $\mathbb{R}^d$ into at most $c_1(mt)^{4d}$ cells such that each cell has the property that at most $N/(mt)$ surfaces from Σ crosses it. We note that $c_1 = c_1(d,t)$ is defined in Lemma 2.2. By the pigeonhole principle, there is a cell Δ in the decomposition such that $|\Delta \cap P| \geq N/(c_1(mt)^{4d})$. Set $P_1 = \Delta \cap P$.

Let $P_2 \subset P \setminus P_1$ such that each point in P_2 gives rise to mt surfaces that do not cross Δ. More precisely,

$$P_2 = \{p \in P \setminus P_1 : Z_{p,i,j} \text{ does not cross } \Delta, \forall i,j\}.$$

Notice that

$$|P_2| \geq N - \frac{N}{mt} - \frac{N}{c_1(mt)^{4d}} \geq \frac{N}{4}.$$

We fix a point $p_0 \in P_1$. Then for each $q \in P_2$, let $\sigma(q) \in \{-1, 0, +1\}^{mt}$ be the sign pattern of the (mt)-tuple $(f_{1,1}(p_0, q), f_{1,2}(p_0, q), ..., f_{m,t}(p_0, q))$. By Theorem 2.1, there are at most $\left(\frac{50mt^2}{d}\right)^d$ distinct sign vectors σ. By the pigeonhole principle, there is a subset $P_3 \subset P_2$ such that

$$|P_3| \geq \frac{|P_2|}{(50/d)^d m^d t^{2d}},$$

and for any two points $q, q' \in P_3$, we have $\sigma(q) = \sigma(q')$. Therefore, for any $p, p' \in P_1$ and $q, q' \in P_3$, $(p, q) \in E_i$ if and only if $(p', q') \in E_i$, for all $i \in \{1, ..., m\}$.

Let $c_4 = c_4(d, t)$ be sufficiently large such that $|P_1|, |P_3| \geq \frac{N}{c_4 m^{4d}}$. By the induction hypothesis, we can find subsets $P_4 \subset P_1, P_5 \subset P_3$, such that

$$|P_4|, |P_5| \geq \left(\frac{N}{c_4 m^{4d}}\right)^{\frac{1}{c_3 \log m}} \geq \frac{N^{\frac{1}{c_3 \log m}}}{2},$$

where $c_3 = c_3(d, t)$ is sufficiently large, and there is an ordering on P_4 (and on P_5) such that each E_i is transitive on P_4 (and on P_5). Set $P' = P_4 \cup P_5$, which implies $|P'| \geq N^{\frac{1}{c_3 \log m}}$. We will show that P' has the desired properties. Let π and π' be the orderings on P_4 and P_5 respectively, such that E_i is transitive on P_4 and on P_5, for every $i \in \{1, ..., m\}$. We order the elements in $P' = \{p_1, ..., p_{|P'|}\}$ by using π and π', such that all elements in P_5 comes after all elements in P_4.

In order to show that E_i is transitive on P', it suffices to examine triples going across P_4 and P_5. Let $p_{j_1}, p_{j_2} \in P_4$ and $p_{j_3} \in P_5$ such that $j_1 < j_2 < j_3$. By construction of P_4 and P_5, if $(p_{j_1}, p_{j_2}), (p_{j_2}, p_{j_3}) \in E_i$, then we have $(p_{i_1}, p_{i_3}) \in E_i$. Likewise, suppose $p_{j_1} \in P_4$ and $p_{j_2}, p_{j_3} \in P_5$. Then again by construction of P_4 and P_5, if $(p_{j_1}, p_{j_2}), (p_{j_2}, p_{j_3}) \in E_i$, then we have $(p_{i_1}, p_{i_3}) \in E_i$. Hence E_i is transitive on P', for all $i \in \{1, ..., m\}$, and this completes the proof.

□

By combining the two previous results, we have the following.

Lemma 3.3. *For $m \geq 2$ and $d, t \geq 1$, let P be a set of N points in $\mathbb{R}^d$ equipped with (symmetric) semi-algebraic relations $E_1, ..., E_m \subset \binom{P}{2}$, where each E_i has complexity at most t. If graph $G_i = (P, E_i)$ has clique number ω_i, then there is a subset $P' \subset P$ of size $\frac{N^{1/(c_3 \log m)}}{\omega_1 \cdots \omega_m}$, where $c_3 = c_3(d, t)$ is defined above, such that $\binom{P'}{2} \cap E_i = \emptyset$ for all i.*

Proof. By applying Lemma 3.2, we obtain a subset $P_1 \subset P$ of size $N^{\frac{1}{c_3 \log m}}$, and an ordering on P_1 such that E_i is transitive on P_1 for all i. Then by an m-fold application of Theorem 3.1, the statement follows.

□

4 Proof of Theorem 1.4

Let P be a point set in $\mathbb{R}^d$ and let $E \subset \binom{P}{3}$ be a semi-algebraic relation on P. We say that (P, E) is $K_s^{(3)}$-free if every collection of s points in P contains a triple not in E. Suppose we have ℓ

disjoint subsets $P_1, ..., P_\ell \subset P$. For $1 \leq i_1 < i_2 < i_3 \leq \ell$, we say that the triple $(P_{i_1}, P_{i_2}, P_{i_3})$ is *homogeneous* if $(p_1, p_2, p_3) \in E$ for all $p_1 \in P_{i_1}, p_2 \in P_{i_2}, p_3 \in P_{i_3}$, or $(p_1, p_2, p_3) \notin E$ for all $p_1 \in P_{i_1}, p_2 \in P_{i_2}, p_3 \in P_{i_3}$. For $p_1, p_2 \in P_1 \cup \cdots \cup P_\ell$ and $i \in \{1, ..., \ell\}$, we say that the triple (p_1, p_2, i) is *good*, if $(p_1, p_2, p_3) \in E$ for all $p_3 \in P_i$, or $(p_1, p_2, p_3) \notin E$ for all $p_3 \in P_i$. We say that the triple (p_1, p_2, i) is *bad* if (p_1, p_2, i) is not good and $p_1, p_2 \notin P_i$.

Lemma 4.1. *Let P be a set of N points in $\mathbb{R}^d$ and let $E \subset \binom{P}{3}$ be a (symmetric) semi-algebraic relation on P such that E has complexity at most t. Then for $r = \frac{N^{1/(30d)}}{tc_2}$, where c_2 is defined in Lemma 2.3, there are disjoint subsets $P_1, ..., P_r \subset P$ such that*

1. *$|P_i| \geq \frac{N^{1/(30d)}}{tc_2}$,*
2. *all triples $(P_{i_1}, P_{i_2}, P_{i_3})$, $1 \leq i_1 < i_2 < i_3 \leq r$, are homogeneous, and*
3. *all triples (p, q, i), where $i \in \{1, ..., r\}$ and $p, q \in (P_1 \cup \cdots \cup P_r) \setminus P_i$, are good.*

Proof. We can assume that $N > (tc_2)^{30d}$, since otherwise the statement is trivial. Since E is semi-algebraic with complexity t, there are polynomials $f_1, ..., f_t$ of degree at most t, and a Boolean function Φ such that

$$(p_1, p_2, p_3) \in E \quad \Leftrightarrow \quad \Phi(f_1(p_1, p_2, p_3) \geq 0, ..., f_t(p_1, p_2, p_3) \geq 0) = 1.$$

For each $p, q \in P$ and $i \in \{1, ..., t\}$, we define the surface $Z_{p,q,i} = \{x \in \mathbb{R}^d : f_i(p, q, x) = 0\}$. Then we set

$$\Sigma = \{Z_{p,q,i} : p, q \in P, 1 \leq i \leq t\}.$$

Thus we have $|\Sigma| = N^2 t$. Next we apply Lemma 2.3 to P and Σ with parameter $\ell = \sqrt{N}$, and obtain subsets $Q_1, ..., Q_\ell$ and cells $\Delta_1, ..., \Delta_\ell$, such that $Q_i \subset \Delta_i$, $|Q_i| = \lfloor \sqrt{N}/4 \rfloor$, and each surface in Σ crosses at most $c_2 N^{1/2 - 1/(4d)}$ cells Δ_i. We note that $c_2 = c_2(d, t)$ is defined in Lemma 2.3 and $\sqrt{N} \geq \log(tN^2)$. Set $Q = Q_1 \cup \cdots \cup Q_\ell$. Each pair $(p, q) \in \binom{Q}{2}$ gives rise to $2t$ surfaces in Σ. By Lemma 2.3, these $2t$ surfaces cross in total at most $2tc_2 N^{1/2 - 1/(4d)}$ cells Δ_i. Hence there are at most $2tc_2 N^{5/2 - 1/(4d)}$ bad triples of the form (p, q, i), where $i \in \{1, ..., \sqrt{N}\}$ and $p, q \in Q \setminus Q_i$. Moreover, there are at most $2tc_2 N^{2 - 1/(4d)}$ bad triples (p, q, i), where both p and q lie in the same part Q_j and $j \neq i$.

We uniformly at random pick $r = \frac{N^{1/(30d)}}{tc_2}$ subsets (parts) from the collection $\{Q_1, ..., Q_\ell\}$, and r vertices from each of the subsets that were picked. For a bad triple (p, q, i) with p and q in distinct subsets, the probability that (p, q, i) survives is at most

$$\left(\frac{r}{\sqrt{N}}\right)^3 \left(\frac{r}{\sqrt{N}/4}\right)^2 = \frac{16}{(tc_2)^5} N^{1/(6d) - 5/2}.$$

For a bad triple (p, q, i) with p, q in the same subset Q_j, where $j \neq i$, the probability that the triple (p, q, i) survives is at most

$$\left(\frac{r}{\sqrt{N}}\right)^2 \left(\frac{r}{\sqrt{N}/4}\right)^2 = \frac{16}{(tc_2)^4} N^{2/(15d) - 2}.$$

Therefore, the expected number of bad triples in our random subset is at most

$$\left(\frac{16}{(tc_2)^5}N^{1/(6d)-5/2}\right)\left(tc_2N^{5/2-1/(4d)}\right)+\left(\frac{16}{(tc_2)^4}N^{2/(15d)-2}\right)\left(tc_2N^{2-1/(4d)}\right)<1.$$

Hence we can find disjoint subsets $P_1,...,P_r$, such that $|P_i|\geq r=\frac{N^{1/(30d)}}{tc_2}$, and there are no bad triples (p,q,i), where $i\in\{1,...,r\}$ and $p,q\in(P_1\cup\cdots\cup P_r)\setminus P_i$.

It remains to show that every triple $(P_{i_1},P_{i_2},P_{i_3})$ is homogeneous for $1\leq i_1<i_2<i_3\leq r$. Let $p_1,\in P_{i_1},p_2\in P_{i_2},p_3\in P_{i_3}$ and suppose $(p_1,p_2,p_3)\in E$. Then for any choice $q_1,\in P_{i_1},q_2\in P_{i_2},q_3\in P_{i_3}$, we also have $(q_1,q_2,q_3)\in E$. Indeed, since the triple (p_1,p_2,i_3) is good, this implies that $(p_1,p_2,q_3)\in E$. Since the triple (p_1,q_3,i_2) is also good, we have $(p_1,q_2,q_3)\in E$. Finally since (q_2,q_3,i_1) is good, we have $(q_1,q_2,q_3)\in E$. Likewise, if $(p_1,p_2,p_3)\not\in E$, then $(q_1,q_2,q_3)\not\in E$ for any $q_1,\in P_{i_1},q_2\in P_{i_2},q_3\in P_{i_3}$. □

We are finally ready to prove Theorem 1.4, which follows immediately from the following theorem.

Theorem 4.2. *Let P be a set of N points in $\mathbb{R}^d$ and let $E\subset\binom{P}{3}$ be a (symmetric) semi-algebraic relation on P such that E has complexity at most t. If (P,E) is $K_s^{(3)}$-free, then there exists a subset $P'\subset P$ such that $\binom{P'}{3}\cap E=\emptyset$ and*

$$|P'|\geq 2^{\frac{(\log\log N)^2}{c^s\log\log\log N}},$$

where $c=c(d,t)$.

Proof. The proof is by induction on N and s. The base cases are $s=3$ or $N\leq(tc_2)^{30d}$, where c_2 is defined in Lemma 2.3. When $N\leq(tc_2)^{30d}$, the statement holds trivially for sufficiently large $c=c(d,t)$. If $s=3$, then again the statement follows immediately by taking $P'=P$.

Now assume that the statement holds if $s'\leq s,N'\leq N$ and not both inequalities are equalities. We apply Lemma 4.1 to (P,E) and obtain disjoint subsets $P_1,...,P_r$, where $r=\frac{N^{1/(30d)}}{tc_2}$, such that $|P_i|\geq\frac{N^{1/(30d)}}{tc_2}$, every triple of parts $(P_{i_1},P_{i_2},P_{i_3})$ is homogeneous, and every triple (p,q,i) is good where $i\in\{1,...,r\}$ and $p,q\in(P_1\cup\cdots\cup P_r)\setminus P_i$.

Let P_0 be the set of $\frac{N^{1/(30d)}}{tc_2}$ points obtained by selecting one point from each P_i. Since (P_0,E) is $K_s^{(3)}$-free, we can apply the induction hypothesis on P_0, and find a set of indices $I=\{i_1,...,i_m\}$ such that

$$\log|I|\geq\frac{\left(\log\log\frac{N^{1/(30d)}}{tc_2}\right)^2}{c^s\log\log\log\frac{N^{1/(30d)}}{tc_2}}\geq(1/2)\log\log N,$$

and for every triple $i_1<i_2<i_3$ in I all triples with one point in each P_{i_j} does not satisfy E. Hence we have $m=\sqrt{\log N}$, and let $Q_j=P_{i_j}$ for $1\leq j\leq m$.

For each subset Q_i, we define binary semi-algebraic relations $E_{i,j}\subset\binom{Q_i}{2}$, where $j\neq i$, as follows. Since $E\subset\binom{P}{3}$ is semi-algebraic with complexity t, there are t polynomials $f_1,...,f_t$ of degree at most t, and a Boolean function Φ such that $(p_1,p_2,p_3)\in E$ if and only if

$$\Phi(f_1(p_1,p_2,p_3)\geq 0,...,f_t(p_1,p_2,p_3)\geq 0)=1.$$

Fix a point $q_0\in Q_j$, where $j\neq i$. Then for $p_1,p_2\in Q_i$, we have $(p_1,p_2)\in E_{i,j}$ if and only if

$$\Phi(f_1(p_1,p_2,q_0) \geq 0, ..., f_t(p_1,p_2,q_0) \geq 0) = 1.$$

Suppose there are $2^{(\log N)^{1/4}}$ vertices in Q_i that induces a clique in the graph $G_{i,j} = (Q_i, E_{i,j})$. Then these vertices would induce a $K_{s-1}^{(3)}$-free subset in the original (hypergraph) (P,E). By the induction hypothesis, we can find a subset $Q_i' \subset Q_i$ such that

$$|Q_i'| \geq 2^{\frac{((1/4)\log\log N)^2}{c^{s-1}\log\log\log N}} \geq 2^{\frac{(\log\log N)^2}{c^s \log\log\log N}},$$

for sufficiently large c, such that $\binom{Q_i'}{3} \cap E = \emptyset$ and we are done. Hence we can assume that each graph $G_{i,j} = (Q_i, E_{i,j})$ has clique number at most $2^{(\log n)^{1/4}}$. By applying Lemma 3.3 to each Q_i, where Q_i is equipped with $m-1$ semi-algebraic relations $E_{i,j}$, $j \neq i$, we can find subsets $T_i \subset Q_i$ such that

$$|T_i| \geq \frac{|Q_i|^{1/(c_3 \log m)}}{2^{(\log N)^{1/4}\sqrt{\log N}}} = \frac{2^{\frac{\log N}{30dc_3\log(\sqrt{\log N})}}}{2^{(\log N)^{3/4}}} \geq 2^{\frac{\log N}{c_5 \log\log N}},$$

where $c_5 = c_5(d,t)$, and $\binom{T_i}{2} \cap E_j = \emptyset$ for all $j \neq i$. Therefore, we now have subsets $T_1, ..., T_m$, such that

1. $m = \sqrt{\log N}$,

2. for any triple $(T_{i_1}, T_{i_2}, T_{i_3})$, $1 \leq i_1 < i_2 < i_3 \leq m$, every triple with one vertex in each T_{i_j} is not in E,

3. for any pair (T_{i_1}, T_{i_2}), $1 \leq i_1 < i_2 \leq m$, every triple with two vertices T_{i_1} and one vertex in T_{i_2} is not in E, and every triple with two vertices T_{i_2} and one vertex in T_{i_1} is also not in E.

By applying the induction hypothesis to each (T_i, E), we obtain a collection of subsets $U_i \subset T_i$ such that

$$\log|U_i| \geq \frac{\left(\log\left(\frac{\log N}{c_5 \log\log N}\right)\right)^2}{c^s \log\log\left(\frac{\log N}{c_5\log\log N}\right)} \geq \frac{(\log\log N - \log(c_5 \log\log N))^2}{c^s \log\log\log N},$$

and $\binom{U_i}{3} \cap E = \emptyset$. Let $P' = \bigcup_{i=1}^{m} U_i$. Then by above we have $\binom{P'}{3} \cap E = \emptyset$ and

$$\begin{aligned}
\log|P'| &\geq \frac{(\log\log N - \log(c_5\log\log N))^2}{c^s \log\log\log N} + \frac{1}{2}\log\log N \\
&\geq \frac{(\log\log N)^2 - 2(\log\log N)\log(c_5\log\log N) + (\log(c_5\log\log N))^2}{c^s\log\log\log N} + \frac{1}{2}\log\log N \\
&\geq \frac{(\log\log N)^2}{c^s \log\log\log N},
\end{aligned}$$

for sufficiently large $c = c(d,t)$.

□

5 Application: One-sided hyperplanes

Let us consider a finite set H of hyperplanes in $\mathbb{R}^d$ in general position, that is, every d members in H intersect at a distinct point. Let $OSH_d(s,n)$ denote the smallest integer N such that every set H of N hyperplanes in $\mathbb{R}^d$ in general position contains s members H_1 such that the vertex set of the arrangement of H_1 lies above the $x_d = 0$ hyperplane, or contains n members H_2 such that the vertex set of the arrangement of H_2 lies below the $x_d = 0$ hyperplane.

In 1992, Matoušek and Welzl [29] observed that $OSH_2(s,n) = (s-1)(n-1)+1$. Dujmović and Langerman [15] used the existence of $OSH_d(n,n)$ to prove a ham-sandwich cut theorem for hyperplanes. Again by adapting the Erdős-Rado argument, Conlon et al. [13] showed that for $d \geq 3$,

$$OSH_d(s,n) \leq \mathrm{twr}_{d-1}(c_6 sn\log n), \tag{4}$$

where c_6 is a constant that depends only on d. See Eliáš and Matoušek [16] for more related results, including lower bound constructions.

Since each hyperplane $h_i \in H$ is specified by the linear equation

$$a_{i,1}x_1 + \cdots + a_{i,d}x_d = b_i,$$

we can represent $h_i \in H$ by the point $h_i^* \in \mathbb{R}^{d+1}$ where $h_i^* = (a_{i,1},...,a_{i,d},b_i)$ and let $P = \{h_i^* : h_i \in H\}$. Then we define a relation $E \subset \binom{P}{d}$ such that $(h_{i_1}^*,...,h_{i_d}^*) \in E$ if and only if $h_{i_1} \cap \cdots \cap h_{i_d}$ lies above the hyperplane $x_d = 0$ (i.e. the d-th coordinate of the intersection point is positive). Clearly, E is a semi-algebraic relation with complexity at most $t = t(d)$. Therefore, as an application of Theorem 1.4 and Corollary 1.5, we make the following improvement on (4).

Theorem 5.1. *For fixed $s \geq 4$, we have $OSH_3(s,n) \leq 2^{n^{o(1)}}$. For fixed $d \geq 4$ and $s \geq d+1$, we have*

$$OSH_d(s,n) \leq \mathrm{twr}_{d-1}(n^{o(1)}).$$

6 Monochromatic triangles

In this section, we will prove Theorem 1.6.

Proof of Theorem 1.6. We proceed by induction on m. The base case when $m = 1$ is trivial. Now assume that the statement holds for $m' < m$. Set $N = 2^{cm\log\log m}$, where $c = c(d,t)$ will be determined later, and let $E_1,...,E_m \subset \binom{P}{2}$ be semi-algebraic relations on P such that $\binom{P}{2} = E_1 \cup \cdots \cup E_m$, and each E_i has complexity at most t. For sake of contradiction, suppose P does not contain three points such that every pair of them is in E_i for some fixed i.

For each relation E_i, there are t polynomials $f_{i,1},...,f_{i,t}$ of degree at most t, and a Boolean function Φ_i such that

$$(p,q) \in E_i \quad \Leftrightarrow \quad \Phi_i(f_{i,1}(p,q) \geq 0,...,f_{i,t}(p,q) \geq 0) = 1.$$

For $1 \leq i \leq m, 1 \leq j \leq t, p \in P$, we define the surface $Z_{i,j,p} = \{x \in \mathbb{R}^d : f_{i,j}(p,x) = 0\}$, and let

$$\Sigma = \{Z_{i,j,p} : 1 \leq i \leq m, 1 \leq j \leq t, p \in P\}.$$

Hence $|\Sigma| = mtN$. We apply Lemma 2.2 to Σ with parameter $r = 2tm$, and decompose $\mathbb{R}^d$ into $c_1(2tm)^{2d}$ regions Δ_i, where $c_1 = c_1(t,d)$ is defined in Lemma 2.2, such that each region Δ_i is crossed by at most $tmN/r = N/2$ members in Σ. By the pigeonhole principle, there is a region $\Delta \subset \mathbb{R}^d$, such that $|\Delta \cap P| \geq \frac{N}{c_1(2tm)^{2d}}$, and at most $N/2$ members in Σ crosses Δ. Let P_1 be a set of exactly $\left\lfloor \frac{N}{c_1(2tm)^{2d}} \right\rfloor$ points in $P \cap \Delta$, and let P_2 be the set of points in $P \setminus P_1$ that does not give rise to a surface that crosses Δ. Hence

$$|P_2| \geq N - \frac{N}{c_1(2tm)^{2d}} - \frac{N}{2} \geq \frac{N}{4}.$$

Therefore, each point $p \in P_2$ has the property that $p \times P_1 \subset E_i$ for some fixed i. We define the function $\chi : P_2 \to \{1,...,m\}$, such that $\chi(p) = i$ if and only if $p \times P_1 \subset E_i$. Set $I = \{\chi(p) : p \in P_2\}$ and $m_0 = |I|$, that is, m_0 is the number of distinct relations (colors) between the sets P_1 and P_2. Now the proof falls into 2 cases.

Case 1. Suppose $m_0 > \log m$. Every pair of points in P_1 is in E_i where $i \in \{1,...,m\} \setminus I$. By the induction hypothesis, we have

$$\frac{2^{cm\log\log m}}{c_1(2tm)^{2d}} \leq |P_1| \leq 2^{c(m-m_0)\log\log m}.$$

Hence

$$cm_0 \log\log m \leq \log(c_1(2tm)^{2d}) \leq 2d\log(c_1 2tm),$$

which implies

$$m_0 \leq \frac{2d\log(c_1 2tm)}{c\log\log m},$$

and we have a contradiction for sufficiently large $c = c(d,t)$.

Case 2. Suppose $m_0 \leq \log m$. By the pigeonhole principle, there is a subset $P_3 \subset P_2$, such that $|P_3| \geq \frac{N}{4m_0}$ and $P_1 \times P_3 \subset E_i$ for some fixed i. Hence every pair of points $p,q \in P_3$ satisfies $(p,q) \notin E_i$, for some fixed i. By the induction hypothesis, we have

$$\frac{2^{cm\log\log m}}{4m_0} \leq |P_3| \leq 2^{c(m-1)\log\log m}.$$

Therefore

$$c\log\log m \leq \log(4m_0) \leq \log(4\log(m)),$$

which is a contradiction since c is sufficiently large. This completes the proof of Theorem 1.6.

□

6.1 Lower bound construction and Schur numbers

Before we prove Theorem 1.7, let us recall a classic Theorem of Schur [31] which is considered to be one of the earliest applications of Ramsey Theory. A subset of numbers $P \subset \mathbb{R}$ is said to be *sum-free* if for any two (not necessarily distinct) elements $x, y \in P$, their sum $x + y$ is not in P.

The Schur number $S(m)$ is defined to be the maximum integer N for which the integers $\{1, ..., N\}$ can be partitioned into m sum-free sets.

Given a partition $\{1, ..., N\} = P_1 \cup \cdots \cup P_m$ into m parts such that P_i is sum-free, we can define an m-coloring on the edges on a complete $(N+1)$-vertex graph which does not contain a monochromatic triangle as follows. Let $V = \{1, ..., N+1\}$ be the vertex set, and we define the coloring $\chi : \binom{V}{2} \rightarrow m$ by $\chi(x, y) = i$ iff $|x - y| \in P_i$. Now suppose for sake of contradiction there are vertices x, y, z that induces a monochromatic triangle, say with color i, such that $x < y < z$. Then we have $y - x, z - y, z - x \in P_i$ and $(y - x) + (z - y) = (z - x)$, which is a contradiction since P_i is sum free. Therefore $S(m) < R_2(\underbrace{3, ..., 3}_{m})$.

Since Schur's original 1916 paper, the lower bound on $S(m)$ has been improved by several authors [2, 1, 21], and the current record of $S(m) \geq \Omega(3.19^m)$ is due to Fredricksen and Sweet [23]. Their lower bound follows by computing $S(6) \geq 538$, and using the recursive formula

$$S(m) \geq c_\ell (2S(\ell) + 1)^{m/\ell},$$

which was established by Abbott and Hanson [1]. Fredricksen and Sweet also computed $S(7) \geq 1680$, which we will use to prove Theorem 1.7.

Lemma 6.1. *For each integer $\ell \geq 1$, there is a set P_ℓ of $(1681)^\ell$ points in $\mathbb{R}$ equipped with semi-algebraic relations $E_1, ..., E_{7\ell} \subset \binom{P_\ell}{2}$, such that*

1. *$E_1 \cup \cdots \cup E_{7\ell} = \binom{P_\ell}{2}$,*
2. *E_i has complexity at most 5000,*
3. *E_i is translation invariant, that is, $(x, y) \in E_i$ iff $(x + C, y + C) \in E_i$, and*
4. *the graph $G_{\ell,i} = (P_\ell, E_i)$ is triangle free for all i.*

Proof. We start be setting $P_1 = \{1, 2, ..., 1681\}$. By [23], there is a partition on $\{1, ..., 1680\} = A_1 \cup \cdots \cup A_7$ into seven parts, such that each A_i is sum-free. For $i \in \{1, ..., 7\}$, we define the relation E_i by

$$(x, y) \in E_i \quad \Leftrightarrow \quad (1 \leq |x - y| \leq 1680) \wedge (|x - y| \in A_i).$$

Since $|A_i| \leq 1680$, E_i has complexity at most 5000. By the arguments above, the graph $G_{1,i} = (P_1, E_i)$ is triangle free for all $i \in \{1, ..., 7\}$.

Having defined $P_{\ell-1}$ and $E_1, ..., E_{7\ell-7}$, we define P_ℓ and $E_{\ell-6}, ..., E_\ell$ as follows. Let $C = C(\ell)$ be a very large constant, say $C > (5000 \cdot \max\{P_{\ell-1}\})^2$. We construct 1681 translated copies of $P_{\ell-1}$, $Q_i = P_{\ell-1} + iC$ for $1 \leq i \leq 1681$, and set $P_\ell = Q_1 \cup \cdots \cup Q_{1681}$. For $1 \leq j \leq 7$, we define the relation $E_{\ell-7+j}$ by

$$(x, y) \in E_{\ell-7+j} \quad \Leftrightarrow \quad (C/2 \leq |x - y| \leq 1682C) \wedge (\exists z \in A_j : ||x - y|/C - z| < 1/1000).$$

Clearly $E_1, ..., E_{7\ell}$ satisfies properties (1), (2), and (3). The fact that $G_{\ell,i} = (P_\ell, E_i)$ is triangle follows from the same argument as above.

□

Theorem 1.7 immediately follows from Lemma 6.1.

7 Concluding remarks

1. We showed that given an N-element point set P in $\mathbb{R}^d$ equipped with a semi-algebraic relation $E \subset \binom{P}{3}$, such that E has complexity at most t and (P, E) is $K_s^{(3)}$-free, then there is a subset $P' \subset P$ such that $|P'| \geq 2^{(\log\log N)^2/(c^s \log\log\log N)}$ and $\binom{P'}{3} \cap E = \emptyset$. In [13], Conlon et al. conjectured that one can find a much larger "independent set". More precisely, they conjectured that there is a constant $\epsilon = \epsilon(d, t, s)$ such that $|P'| \geq N^{\epsilon}$. Perhaps an easier task would be to find a large subset P' such that E is *transitive* on P', that is, there is an ordering on $P' = \{p_1, ..., p_m\}$ such that for $1 \leq i_1 < i_2 < i_3 < i_4 \leq m$, $(p_{i_1}, p_{i_2}, p_{i_3}), (p_{i_2}, p_{i_3}, p_{i_4}) \in E$ implies that $(p_{i_1}, p_{i_2}, p_{i_4}), (p_{i_1}, p_{i_3}, p_{i_4}) \in E$.

2. Off diagonal Ramsey numbers for binary semi-algebraic relations. As mentioned in the introduction, we have $R_2(s, n) \leq O(n^{s-1})$. It would be interesting to see if one could improve this upper bound in our semi-algebraic setting. That is, for fixed integers $t \geq 1$ and $d \geq s \geq 3$, is there an constant $\epsilon = \epsilon(d, t, s)$ such that $R_2^{d,t}(s, n) \leq O(n^{s-1-\epsilon})$? For $d < s$, it is likely that such an improvement can be made using Lemma 2.2.

3. Low complexity version of Schur's Theorem. We say that the subset $P' \subset \{1, ..., N\}$ has complexity t if there are t intervals $I_1, ..., I_t$ such hat $P' = \{1, ..., N\} \cap (I_1 \cup \cdots \cup I_t)$. Let $S_t(m)$ be the minimum integer N such that any partition $\{1, ..., N\} = P_1 \cup \cdots \cup P_m$ into m parts, such that each part has complexity at most t, contains a part P_i that contains two numbers x and y along with their sum $x + y$. By following the proof Theorem 1.6, one can show that $S_t(m) \leq 2^{m \log\log 2t}$.

References

[1] H. L. Abbott and D. Hanson, A problem of Schur and its generalizations, *Acta Arith.* **20** (1972), 175–187.

[2] H. L. Abbott and L. Moser, Sum-free sets of integers, *Acta Arith.* **11** (1966), 392–396.

[3] P. K. Agarwal and J. Erickson, Geometric range searching and its relatives, *Advances in Dicsrete and Computational Geometry* (B. Chazelle, J.E. Goodman, and R. Pollack, eds.), AMS Press, Providence, RI, 1998, 1–56.

[4] M. Ajtai, J. Komlós, and E. Szemerédi, A note on Ramsey numbers, *J. Combin. Theory Ser. A* **29** (1980), 354–360.

[5] N. Alon, J. Pach, R. Pinchasi, R. Radoičić, and M. Sharir, Crossing patterns of semi-algebraic sets, *J. Combin. Theory Ser. A* **111** (2005), 310–326.

[6] S. Basu, R. Pollack, and M. F. Roy, *Algorithms in Real Algebraic Geometry*, 2nd Edition, Algorithms and Computation in Mathematics, 10, Springer-Verlag, Berlin, 2006.

[7] T. Bohman, The triangle-free process, *Adv. Math.* **221** (2009), 1653–1677.

[8] T. Bohman and P. Keevash, The early evolution of the H-free process, *Invent. Math.* **181** (2010), 291–336.

[9] B. Bukh and M. Matoušek, Erdős-Szekeres-type statements: Ramsey function and decidability in dimension 1, to appear in *Duke Math. Journal.*

[10] T. Chan, Optimal partition trees, *In Proc. 26th Ann. ACM Sympos. Comput. Geom.* (2010), 1–10.

[11] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. A singly exponential stratification scheme for real semi-algebraic varieties and its applications. Theor. Comput. Sci. **84** (1991), 77–105.

[12] K. L. Clarkson and P. W. Shor, Applications of random sampling in computational geometry, II, *Discrete Comput. Geom.* **4** (1989), 387–421.

[13] D. Conlon, J. Fox, J. Pach, B. Sudakov, and A. Suk, Ramsey-type results for semi-algebraic relations, *Trans. Amer. Math. Soc.*, to appear.

[14] D. Conlon, J. Fox, and B. Sudakov, Hypergraph Ramsey numbers, *J. Amer. Math. Soc.* **23** (2010), 247–266.

[15] V. Dujmović and S. Langerman, A center transversal theorem for hyperplanes and applications to graph drawing, in Proc. 27th ACM Symposium on Computational Geometry (2011), 117–124.

[16] M. Eliáš and J. Matoušek, Higher-order Erdős-Szekeres theorems, *Advances in Mathematics* **244** (2013), 1–15.

[17] P. Erdős, Some remarks on the theory of graphs, *Bull. Amer. Math. Soc.* **53** (1947), 292–294.

[18] P. Erdős and R. Rado, Combinatorial theorems on classifications of subsets of a given set, *Proc. London Math. Soc.* **3** (1952), 417–439.

[19] P. Erdős, A. Hajnal, and R. Rado, Partition relations for cardinal numbers, *Acta Math. Acad. Sci. Hungar.* **16** (1965), 93–196.

[20] P. Erdős and G. Szekeres, A combinatorial problem in geometry, *Compos. Math.* **2** (1935), 463–470.

[21] G. Exoo, A Lower Bound for Schur Numbers and Multicolor Ramsey Numbers of K_3, *Electronic J. Combinatorics* **1**, R8 (1994), 1–3.

[22] J. Fox, J. Pach, B. Sudakov, and A. Suk, Erdős-Szekeres-type theorems for monotone paths and convex bodies, *Proceedings of the London Mathematical Society* **105** (2012), 953–982.

[23] H. Fredricksen and M. Sweet, Symmetric sum-free partitions and lower bounds for Schur numbers, *Electronic J. Combinatorics* **7**, R32 (2000), 1–9.

[24] R. L. Graham, B. L. Rothschild, and J. H. Spencer, *Ramsey Theory*, 2nd Edition, Wiley, New York, 1990.

[25] J. H. Kim, The Ramsey number $R(3,t)$ has order of magnitude $t^2/\log t$, *Random Structures Algorithms* **7** (1995), 173–207.

[26] V. Koltun, Almost tight upper bounds for vertical decompositions in four dimensions, *J. ACM* **51** (2004), 699-730.

[27] M. Eliáš, J. Matoušek, E. Roldán-Pensado, and Z. Safernová, Lower bounds on geometric Rmasey functions, Arxiv preprint 1307.5157, 2014.

[28] J. Matoušek, Efficient partition trees, *Discrete Comput. Geom.* **8** (1992), 315–334.

[29] J. Matoušek and E. Welzl, Good splitters for counting points in triangles, *J. Algorithms* **13** (1992), 307–319.

[30] J. Milnor, On the Betti numbers of real varieties, *Proc. Amer. Math. Soc.* **15** (1964), 275–280.

[31] I. Schur, Über die Kongruenz $x^m + y^m = z^m \mod p$, *Jahresber. Deutch. Math. Verein.* **25** (1916), 114–117.

[32] M. J. Steele, Variations on the monotone subsequence theme of Erdős and Szekeres. In *D. Aldous et al., eitors, Discrete Probability and Algorithms, IMA Volumes in Mathematics and its Applications* **72** (1995), Springer, Berlin etc., 111–131.

[33] A. Suk, A note on order-type homogeneous point sets, *Mathematika* **60** (2014), 37–42.

[34] R. Thom, Sur l'homologie des variétés algébriques réelles, in Differential and Combinatorial Topology (A Symposium in Honor of Marston Morse), Princeton Univ. Press, Princeton, N.J., 1965, 255–265.

Convex Hulls under Uncertainty

Pankaj K. Agarwal[*] Sariel Har-Peled[†] Subhash Suri[‡] Hakan Yıldız[§]
Wuzhou Zhang[¶]

June 26, 2014

Abstract

We study the convex-hull problem in a probabilistic setting, motivated by the need to handle data uncertainty inherent in many applications, including sensor databases, location-based services and computer vision. In our framework, the uncertainty of each input site is described by a probability distribution over a finite number of possible locations including a ***null*** location to account for non-existence of the point. Our results include both exact and approximation algorithms for computing the probability of a query point lying inside the convex hull of the input, time-space tradeoffs for the membership queries, a connection between Tukey depth and membership queries, as well as a new notion of β-hull that may be a useful representation of uncertain hulls.

1. Introduction

The convex hull of a set of points is a fundamental structure in mathematics and computational geometry, with wide-ranging applications in computer graphics, image processing, pattern recognition, robotics, combinatorics, and statistics. Worst-case optimal as well as output-sensitive algorithms are known for computing the convex hull; see the survey [Sei04] for an overview of known results.

In many applications, such as sensor databases, location-based services or computer vision, the location and sometimes even the existence of the data is uncertain, but statistical information can be used as a probability distribution guide for data. This raises the natural computational question: what is a robust and useful convex hull representation for such an uncertain input, and how well can we compute it? We explore this problem under two simple models in which both the location and the existence (presence) of each point is described probabilistically, and study basic questions such as what is the probability of a query point lying inside the convex hull, or what does the probability distribution of the convex hull over the space look like.

[*] Department of Computer Science; Duke University; Durham, NC, 27708, USA; pankaj@cs.duke.edu <http://www.cs.duke.edu/>; http://www.cs.duke.edu/\protect\unhbox\voidb@x\penalty\@M\{}pankaj/.

[†] Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA;sariel@uiuc.edu; http://sarielhp.org/ .

[‡] Department of Computer Science; University of California; Santa Barbara; CA, 93106, USA; suri@cs.ucsb.edu; http://www.cs.ucsb.edu/\protect\unhbox\voidb@x\penalty\@M\{}suri.

[§] Department of Computer Science; University of California; Santa Barbara; CA, 93106, USA; hakan@cs.ucsb.edu; http://www.cs.ucsb.edu/\protect\unhbox\voidb@x\penalty\@M\{}hakan.

[¶] Department of Computer Science; Duke University; Durham, NC, 27708, USA; wuzhou@cs.duke.edu <http://www.cs.duke.edu/>; http://www.cs.duke.edu/\protect\unhbox\voidb@x\penalty\@M\{}wuzhou/.

Uncertainty models. We focus on two models of uncertainty: unipoint and multipoint. In the ***unipoint model***, each input point has a fixed location but it only exists probabilistically. Specifically, the input $\mathcal{P}$ is a set of pairs $\{(p_1, \gamma_1), \ldots, (p_n, \gamma_n)\}$ where each p_i is a point in $\mathbb{R}^d$ and each γ_i is a real number in the range $(0, 1]$ denoting the probability of p_i's existence. The existence probabilities of different points are independent; $P = \{p_1, \ldots, p_n\}$ denotes the set of sites in $\mathcal{P}$.

In the ***multipoint model***, each point probabilistically exists at one of multiple possible sites. Specifically, $\mathcal{P}$ is a set of pairs $\{(P_1, \Gamma_1), \ldots, (P_m, \Gamma_m)\}$ where each P_i is a set of n_i points and each Γ_i is a set of n_i real values in the range $(0, 1]$. The set $P_i = \{p_i^1, \ldots, p_i^{n_i}\}$ describes the possible sites for the ith point of $\mathcal{P}$ and the set $\Gamma_i = \{\gamma_i^1, \ldots, \gamma_i^{n_i}\}$ describes the associated probability distribution. The probabilities γ_i^j correspond to disjoint events and therefore sum to at most 1. By allowing the sum to be less than one, this model also accounts for the possibility of the point not existing (i.e. the ***null*** location)—thus, the multipoint model generalizes the unipoint model. In the multipoint model, $P = \bigcup_{i=1}^m P_i$ refers to the set of all sites and $n = |P|$.

Our results. The main results of our paper can be summarized as follows.

(A) We show (in Section 2) that the membership probability of a query point $q \in \mathbb{R}^d$, namely, the probability of q being inside the convex hull of $\mathcal{P}$, can be computed in $O(n \log n)$ time for $d = 2$. For $d \geq 3$, assuming the input and the query point are in general position, the membership probability can be computed in $O(n^d)$ time. The results hold for both unipoint and multipoint models.

(B) Next we describe two algorithms (in Section 3) to preprocess $\mathcal{P}$ into a data structure so that for a query point its membership probability in $\mathcal{P}$ can be answered quickly. The first algorithm constructs a ***probability map*** $\mathbb{M}(\mathcal{P})$, a partition of $\mathbb{R}^d$ into convex cells, so that all points in a single cell have the same membership probability. We show that $\mathbb{M}(\mathcal{P})$ has size $\Theta(n^{d^2})$, and for $d = 2$ it can be computed in optimal $O(n^4)$ time. The second one is a sampling-based Monte Carlo algorithm for constructing a near-linear-size data structure that can approximate the membership probability with high likelihood in sublinear time for any fixed dimension.

(C) We show (in Section 4) a connection between the membership probability and the Tukey depth, which can be used to approximate cells of high membership probabilities. For $d = 2$, this relationship also leads to an efficient data structure.

(D) Finally, we introduce the notion of ***β-hull*** (in Section 5) as another approximate representation for uncertain convex hulls in the multipoint model: a convex set C is called ***β-dense*** for $\mathcal{P}$, for $\beta \in [0, 1]$, if C contains at least β fraction of each uncertain point. The β-hull of $\mathcal{P}$ is the intersection of all β-dense sets for $\mathcal{P}$. We show that for $d = 2$, the β-hull of $\mathcal{P}$ can be computed in $O(n \log^3 n)$ time.

Related work. There is extensive and ongoing research in the database community on uncertain data; see [DRS09] for a survey. In the computational geometry community, the early work relied on deterministic models for uncertainty (see e.g. [LÖ9]), but more recently probabilistic models of uncertainty, which are closer to the models used in statistics and machine learning, have been explored [ACTY09, AAH+13, Phi09, KCS11a, KCS11b, SVY13]. The convex-hull problem over uncertain data has received some attention very recently. Suri *et al.* [SVY13] showed that the problem of computing the most likely convex hull of a point set in the multipoint model is NP-hard. Even in the unipoint model, the problem is NP-hard for $d \geq 3$. They also presented an $O(n^3)$-time algorithm for computing the most likely convex hull under the unipoint model in $\mathbb{R}^2$. Zhao *et al.* [ZYN12] investigated the problem of computing the probability of each uncertain point lying on the convex hull, where they aimed to return the set of (uncertain) input points whose probabilities of being on the convex hull are at least some threshold. Jørgensen *et al.* [JLP11] showed that the distribution of properties, such as areas or perimeters, of the convex hull of $\mathcal{P}$ may have $\Omega(\Pi_{i=1}^{m} n_i)$ complexity if all the sites lie on or near a circle.

2. Computing the Membership Probability

For simplicity, we describe our algorithms under the unipoint model, and then discuss their extension to the multipoint model. We begin with the 2D case.

2.1. The two-dimensional case

Let $\mathcal{P} = \{(p_1, \gamma_1), \ldots, (p_n, \gamma_n)\}$ be a set of n uncertain points in $\mathbb{R}^2$ under the unipoint model. Recall that $P = \{p_1, \ldots, p_n\}$ is the set of all sites of $\mathcal{P}$. For simplicity of description, we assume that the sites are in general position, i.e., no two share coordinates and no three are collinear. A subset $B \subseteq P$ is the outcome of a probabilistic experiment with probability

$$\gamma(B) = \prod_{p_i \in B} \gamma_i \times \prod_{p_i \notin B} \overline{\gamma_i},$$

where $\overline{\gamma_i}$ is the complementary probability $1 - \gamma_i$. By definition, for a point q, the ***probability*** of q to lie in the convex-hull of B is

$$\mu(q) = \sum_{B \subseteq P \mid q \in \textsc{ch}(B)} \gamma(B),$$

where $\textsc{ch}(B)$ is the convex hull of B. This unfortunately involves an exponential number of terms. However, observe that for a subset $B \subseteq P$, the point q is ***outside*** $\textsc{ch}(B)$, if and only if q is a vertex of the convex hull $\textsc{ch}(B \cup \{q\})$. So, let $C = \textsc{ch}(B \cup \{q\})$, and V be the set of vertices of C. Then, we have that $\mu(q) = 1 - \Pr[q \in V]$.

If $B = \emptyset$, then clearly $C = \{q\}$ and $q \in V$. Otherwise, $|V| \geq 2$ and $q \in V$ implies that q is an endpoint of exactly two edges on the boundary of C.[1] In this case, the first edge following q in the counter-clockwise order of C is called the ***witness edge*** of q being in V. Hence, $q \in V$ if and only if $B = \emptyset$ or (exclusively) B has a witness edge, i.e.,

$$\Pr\Big[q \in V \Big] = \Pr\Big[B = \emptyset \Big] + \sum_{i=1}^{n} \Pr\Big[qp_i \text{ is the witness edge of } q \notin \textsc{ch}(B) \Big].$$

The first term can be computed in linear time. To compute the ith term in the summation, we observe that qp_i is the witness edge of B if and only if $p_i \in B$ and B contains no sites to the right of the oriented line spanned by the vector $\overrightarrow{qp_i}$, and the corresponding probability is $\gamma_i \cdot \prod_{p_j \in G_i} \overline{\gamma_j}$, where G_i is the set of sites to the right of $\overrightarrow{qp_i}$. This expression can be computed in $O(n)$ time. It follows that one can compute $1 - \mu(q)$, and therefore $\mu(q)$, in $O(n^2)$ time. The computation time can be improved to $O(n \log n)$ as described in the following paragraph.

Improving the running time. The main idea is to compute the witness edge probabilities in radial order around q. We sort all sites in counter-clockwise order around q. Without loss of generality,

[1] If B consists of a single site p_i, then C is the line segment qp_i. In this case, we consider the boundary of C to be a cycle formed by two edges: one going from q to p_i, and one going from p_i back to q.

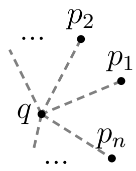


Figure 1: Sites in radial order around q.

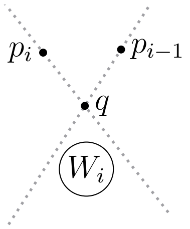


Figure 2: The set W_i.

assume that the circular sequence $p_1, \ldots, p_n$ is the resulting order. (See Figure 1.) We first compute the probability that qp_1 is the witness edge in $O(n)$ time. Then, for increasing values of i from 2 to n, we compute the probability that qp_i is the witness edge by updating the probability for qp_{i-1}, in $O(1)$ amortized time. In particular, let W_i denote the set of sites in the open wedge bounded by the vectors $\overrightarrow{qp_{i-1}}$ and $\overrightarrow{qp_i}$. (See Figure 2.) Notice that $G_i = G_{i-1} \cup \{p_{i-1}\} \setminus W_i$. It follows that the probability for qp_i can be computed by multiplying the probability for qp_{i-1} with $\frac{\gamma_i}{\gamma_{i-1}} \times \frac{\overline{\gamma_{i-1}}}{\prod_{p_j \in W_i} \overline{\gamma_j}}$. The cost of a single update is $O(1)$ amortized because total number multiplications in all the updates is at most $4n$. (Each site affects at most 4 updates.) Finally, notice that we can easily keep track of the set W_i during our radial sweep, as changes to this set follow the same radial order.

Theorem 2.1. *Given a set of n uncertain points in $\mathbb{R}^2$ under the unipoint model, the membership probability of a query point q can be computed in $O(n \log n)$ time.*

2.2. The d-dimensional case

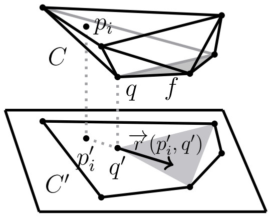
The difficulty in extending the above to higher dimensions is an appropriate generalization of witness edges, which allow us to implicitly sum over exponentially many outcomes without overcounting. Our algorithm requires that all sites, including the query point q, are in general position, i.e., no $k+1$ points of $\mathcal{P} \cup \{q\}$ lie on a $(k-1)$-hyperplane when projected into a subset of k coordinates, where $2 \leq k \leq d$.

Let B be an outcome, $C = \text{CH}(B \cup \{q\})$ its convex hull, and V the vertices of C. Let $\lambda(B \cup \{q\})$ denote the point with the lowest x_d-coordinate in $B \cup \{q\}$. Clearly, if q is $\lambda(B \cup \{q\})$ then $q \in V$; otherwise, we condition the probability based on which point among B is $\lambda(B \cup \{q\})$. Therefore, we can write

$$\Pr\Big[q \in V\Big] = \Pr\Big[q = \lambda(B \cup \{q\})\Big] + \sum_{1 \leq i \leq n} \Pr\Big[p_i = \lambda(B \cup \{q\}) \ \wedge \ q \in V\Big].$$

It is easy to compute the first term. We show below how to compute each term of the summation in $O(n^{d-1})$ time, which gives the desired bound of $O(n^d)$.

Consider an outcome B with $p_i \in B$. Let B', p_i' and q' denote the projections of B, p_i and q respectively on the hyperplane $x_d = 0$, which we identify with $\mathbb{R}^{d-1}$. Let us define $C' = \text{CH}(B' \cup \{q'\}) \subset \mathbb{R}^{d-1}$, and let V' be the vertices of C'.



Let $\overrightarrow{r}(p_i', q')$ denote the open ray emanating from q' in the direction of the vector $\overrightarrow{p_i'q'}$ (that is, this ray is moving "away" from p_i'). A facet f of C is a ***p_i-escaping*** facet for q, if q is a vertex of f and the projection of f on $\mathbb{R}^{d-1}$ intersects $\overrightarrow{r}(p_i', q')$. See the figure on the right. The following lemma is key to our algorithm. The points of C projected into $\partial C'$ form the ***silhouette*** of C.

Lemma 2.2. *(A) If $q' \in V'$ then q is a silhouette vertex of C and vice versa.*

(B) If $p_i \in B$ then q has at most one p_i-escaping facet on C.

(C) The point q is a non-silhouette vertex of the convex-hull C if and only if q has a (single) p_i-escaping facet on C.

Proof: (A) By definition.

(B) If q has a p_i-escaping facet then it is a vertex of the convex-hull C. Consider the union of facets adjacent to q, and observe that the projection of this "tent" can fold over itself in the projection only if q is on the silhouette. Specifically, if q is not on the silhouette then the claim immediately holds.

Otherwise, q is on the silhouette then the open ray $\overrightarrow{r}(p_i', q')$ does not intersect C', and there are no p_i-escaping facets.

(C) Follows immediately from (B), by observing that in this case, the projected "tent", surrounds q', and as such one of the facets must be an escaping facets for p_i. ■

Given a subset of sites $P_\alpha \subseteq P \setminus \{p_i\}$ of size $(d-1)$, define $f(P_\alpha)$ to be the $(d-1)$-dimensional simplex $\textsc{ch}(P_\alpha \cup \{q\})$. Since $p_i = \lambda(B \cup \{q\})$ implies $p_i \in B$, we can use Lemma 2.2 to decompose the ith term as follows:

$$\Pr\Big[p_i = \lambda(B \cup \{q\}) \ \wedge \ q \in V\Big] = \Pr\Big[p_i = \lambda(B \cup \{q\}) \ \wedge \ q' \in V'\Big]$$
$$+ \sum_{\substack{P_\alpha \subseteq P \setminus \{p_i\} \\ |P_\alpha| = (d-1) \\ f(P_\alpha) \text{ is } p_i\text{-escaping for } q}} \Pr\Big[p_i = \lambda(B \cup \{q\}) \ \wedge \ f(P_\alpha) \text{ is a facet of } C\Big].$$

The first term is an instance of the same problem in $(d-1)$ dimensions (for the point q' and the projection of P), and thus is computed recursively. For the second term, we compute the probability that $f(P_\alpha)$ is a facet of C as follows. Let $G_1 \subseteq P$ be the subset of sites which are on the other side of the hyperplane supporting $f(P_\alpha)$ with respect to p_i. Let $G_2 \subseteq P$ be the subset of sites that are below p_i along the x_d-axis. Clearly, $f(P_\alpha)$ is a facet of C (and $p_i = \lambda(B \cup \{q\})$) if and only if all points in P_α and p_i exist in B, and all points in $G_1 \cup G_2$ are absent from B. The corresponding probability can be written as

$$\gamma_i \times \prod_{p_j \in P_\alpha} \gamma_j \times \prod_{p_j \in G_1 \cup G_2} \overline{\gamma_j}\,.$$

This formula is valid only if $P_\alpha \cap G_2 = \emptyset$ and p_i has a lower x_d-coordinate than q; otherwise we set the probability to zero. This expression can be computed in linear time, and the whole summation term can be computed in $O(n^d)$ time. Then, by induction, the computation of the ith term takes $O(n^d)$ time. Notice that the base case of our induction requires computing the probability $\Pr\big[p_i = \lambda(B \cup \{q\}) \ \wedge \ q^{(d-2)} \in V^{(d-2)}\big]$ (where ${}^{(d-2)}$ indicates a projection to $\mathbb{R}^2$). Computing this probability is essentially a two-dimensional membership probability problem on q and P, but is conditioned on the existence of p_i and the non-existence of all sites below p_i along dth axis. Our two dimensional algorithm can be easily adapted to solve this variation in $O(n \log n)$ time as well. (Briefly, we apply the same algorithm but we ignore all points that are below p_i . We later adjust the Finally, we can improve the computation time for the ith term to $O(n^{d-1})$ by considering the facets $f(P_\alpha)$ in radial order. The details can be found in Appendix B.

Remark. The degeneracy of the input is easy to handle in two dimensions, but creates some technical difficulties in higher dimensions that we are currently investigating.

Theorem 2.3. *Let $\mathcal{P}$ be an uncertain set of n points in the unipoint model in $\mathbb{R}^d$ and q be a point. If the input sites and q are in general position, then one can compute the membership probability of q in $O(n^d)$ time, using linear space.*

Extension to the multipoint model. The algorithm extends to the multipoint model easily by modifying the computation of the probability for an edge or facet. Deferring the details to Appendix C, we conclude the following.

Theorem 2.4. *Given an uncertain set $\mathcal{P}$ of n points in the multipoint model in $\mathbb{R}^d$ and a point $q \in \mathbb{R}^d$, we can compute the membership probability of q in $O(n \log n)$ time for $d = 2$, and in $O(n^d)$ time for $d \geq 3$ if input sites and q are in general position.*

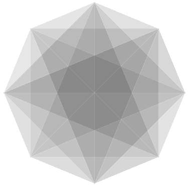
3. Membership Queries

We describe two algorithms – one deterministic and one Monte Carlo – for preprocessing a set of uncertain points for efficient membership-probability queries.

Probability map. The ***probability map*** $\mathbb{M}(\mathcal{P})$ is the subdivision of $\mathbb{R}^d$ into maximal connected regions so that $\mu(q)$ is the same for all query points q in a region. The following lemma gives a tight bound on the size of $\mathbb{M}(\mathcal{P})$.

Lemma 3.1. *The worst-case complexity of the probability map of a set of uncertain points in $\mathbb{R}^d$ is $\Theta(n^{d^2})$, under both the unipoint and the multipoint model, where n is the total number of sites in the input.*

Proof: We prove the result for the unipoint model, as the extension to the multipoint is straightforward. For the upper bound, consider the set H of $O(n^d)$ hyperplanes formed by all d-tuples of points in $\mathcal{P}$. In the arrangement $\mathcal{A}(H)$ formed by these planes, each (open) cell has the same value of $\mu(q)$. This arrangement, which is a refinement of $\mathbb{M}(\mathcal{P})$, has size $O((n^d)^d) = O(n^{d^2})$, establishing the upper bound.



For the lower bound, consider the problem in two dimensions; extension to higher dimensions is straightforward. We choose the sites to be the vertices $p_1, \ldots, p_n$ of a regular n-gon, where each site exists with probability γ, $0 < \gamma < 1$. See the figure on the right. Consider the arrangement $\mathcal{A}$ formed by the line segments p_ip_j, $1 \leq i < j \leq n$, and treat each face as relatively open. If $\mu(f)$ denotes the membership probability for a face f of $\mathcal{A}$, then for any two faces f_1 and f_2 of $\mathcal{A}$, where f_1 bounds f_2 (i.e., $f_1 \subset \partial f_2$), we have $\mu(f_1) \geq \mu(f_2)$, and $\mu(f_1) > \mu(f_2)$ if $\gamma < 1$. Thus, the size of the arrangement $\mathcal{A}$ is also a lower bound on the complexity of $\mathbb{M}(\mathcal{P})$. This proves that the worst-case complexity of $\mathbb{M}(\mathcal{P})$ in $\mathbb{R}^d$ is $\Theta(n^{d^2})$. ■

We can preprocess this arrangement into a point-location data structure, giving us the following result for $d = 2$.

Theorem 3.2. *Let $\mathcal{P}$ be a set of uncertain points in $\mathbb{R}^2$, with a total of n sites. $\mathcal{P}$ can be preprocessed in $O(n^4)$ time into a data structure of size $O(n^4)$ so that for any point $q \in \mathbb{R}^d$, $\mu(q)$ can be computed in $O(\log n)$ time.*

Appendix D describes how to construct the data structure in $O(n^4)$ time.

Remark. For $d \geq 3$, due to our general position assumption, we can compute the membership probability only for d-faces of $\mathbb{M}(\mathcal{P})$, and not for the lower-dimensional faces. In that case, by utilizing a point-location technique in [Cha93], one can build a structure that can report the membership probability of a query point (inside a d-face) in $O(\log n)$ time, with a preprocessing cost of $O(n^{d^2+d})$.

Monte Carlo algorithm. The size of the probability map may be prohibitive even for $d = 2$, so we describe a simple, space-efficient Monte Carlo approach for quickly approximating the membership probability, within absolute error. Fix a parameter $s > 1$, to be specified later. The preprocessing consists of s rounds, where the algorithm creates an outcome A_j of $\mathcal{P}$ in each round j. Each A_j is preprocessed into a data structure so that for a query point $q \in \mathbb{R}^d$, we can determine whether $q \in \textsc{ch}(A_j)$.

For $d \leq 3$, we can build each $\textsc{ch}(A_j)$ explicitly and use linear-size point-location structures with $O(\log n)$ query time. This leads to total preprocessing time $O(sn \log n)$ and space $O(sn)$. For $d \geq 4$, We use the data structure in [MS92] for determining whether $q \in A_j$, for all $1 \leq j \leq s$. For a parameter t such that $n \leq t \leq n^{\lfloor d/2 \rfloor}$ and for any constant $\sigma > 0$, using $O(st^{1+\sigma})$ space and preprocessing, it can compute in $O(\frac{sn}{t^{1/\lfloor d/2 \rfloor}} \log^{2d+1} n)$ time whether $q \in \textsc{ch}(A_j)$ for every j.

Given a query point $q \in \mathbb{R}^d$, we check for membership in all $\textsc{ch}(A_j)$, and if it lies in k of them, we return $\widehat{\mu}(q) = k/s$ as our estimate of $\mu(q)$. Thus, the query time is $O(\frac{sn}{t^{1/\lfloor d/2 \rfloor}} \log^{2d+1} n)$ for $d \geq 4$, $O(s \log n)$ for $d = 3$, and $O(\log n + s)$ for $d = 2$ (using fractional cascading).

It remains to determine the value of s so that $|\mu(q) - \widehat{\mu}(q)| \leq \varepsilon$ for all queries q, with probability at least $1 - \delta$. For a fixed q and outcome A_j, let X_i be the random indicator variable, which is 1 if $q \in \textsc{ch}(A_j)$ and 0 otherwise. Since $\mathsf{E}[X_i] = \mu(q)$ and $X_i \in \{0, 1\}$, using a Chernoff-Hoeffding bound on $\widehat{\mu}(q) = k/s = (1/s) \sum_i X_i$, we observe that $\Pr[\,|\widehat{\mu}(q) - \mu(q)\,| \geq \varepsilon\,] \leq 2\exp(-2\varepsilon^2 s) \leq \delta'$. By Lemma 3.1, we need to consider $O(n^{d^2})$ distinct queries. If we set $1/\delta' = O(n^{d^2}/\delta)$ and $s = O((1/\varepsilon^2)\log(n/\delta))$, we obtain the following theorem.

Theorem 3.3. *Let $\mathcal{P}$ be a set of uncertain points in $\mathbb{R}^d$ under the multipoint model with a total of n sites, and let $\varepsilon, \delta \in (0, 1)$ be parameters. For $d \geq 4$, $\mathcal{P}$ can be preprocessed, for any constant $\sigma > 0$, in $O((t^{1+\sigma}/\varepsilon^2) \log \frac{n}{\delta})$ time, into a data structure of size $O((t^{1+\sigma}/\varepsilon^2) \log \frac{n}{\delta})$, so that with probability at least $1 - \delta$, for any query point $q \in \mathbb{R}^2$, $\widehat{\mu}(q)$ satisfying $|\mu(q) - \widehat{\mu}(q)\,| \leq \varepsilon$ and $\widehat{\mu}(q) > 0$ can be returned in $O(\frac{n}{t^{1/\lfloor d/2 \rfloor} \varepsilon^2} \log \frac{n}{\delta} \log^{2d+1} n)$ time, where t is a parameter and $n \leq t \leq n^{\lfloor d/2 \rfloor}$. For $d \leq 3$, the preprocessing time and space are $O(\frac{n}{\varepsilon^2} \log\log \frac{n}{\delta} \log n)$ and $O(\frac{n}{\varepsilon^2} \log \frac{n}{\delta})$, respectively. The query time is $O(\frac{1}{\varepsilon^2} \log(\frac{n}{\delta}) \log n)$ (resp. $O(\frac{1}{\varepsilon^2} \log \frac{n}{\delta})$) for $d = 3$ (resp. $d = 2$).*

4. Tukey Depth and Convex Hull

The membership probability is neither a convex nor a continuous function, as suggested by the example in the proof of Lemma 3.1. In this section, we establish a helpful structural property of this function, intuitively showing that the probability stabilizes once we go deep enough into the "region". Specifically, we show a connection between the Tukey depth of a point q with its membership probability; in two dimensions, this also results in an efficient data structure for approximating $\mu(q)$ quickly within a small absolute error.

Estimating $\mu(q)$. Let Q be a set of weighted points in $\mathbb{R}^d$. For a subset $A \subseteq Q$, let $w(A)$ be the total weight of points in A. Then the ***Tukey depth*** of a point $q \in \mathbb{R}^d$ with respect to Q, denoted by $\tau(q, Q)$, is $\min w(Q \cap H)$ where the minimum is taken over all halfspaces H that contain q.[2] If Q is obvious from the context, we use $\tau(q)$ to denote $\tau(q, Q)$. Before bounding $\mu(q)$ in terms of $\tau(q, Q)$, we prove the following lemma.

Lemma 4.1. *Let Q be a finite set of points in $\mathbb{R}^d$. For any $p \in \mathbb{R}^d$, there is a set $\mathcal{S} = \{S_1, \ldots, S_T\}$ of d-simplices formed by Q such that (i) each S_i contains p in its interior; (ii) no pair of them shares a vertex; and (iii) $T \geq \lceil \tau(p, Q)/d \rceil$.*

Proof: As long as $\tau(p, Q) > 0$, $p \in \textsc{ch}(Q)$, and by Carathéodory Theorem [Eck93], there is a d-simplex S with its $d+1$ vertices in Q such that $p \in S$. Remove the vertices of S from Q, and repeat the argument. Let $S_1, \ldots, S_T$ be the resulting simplices. Observe that at most d vertices of S can be in an halfspace passing through p, which implies that the Tukey depth of p drops by at most d after each iteration of this algorithm. Hence $T \geq \lceil \tau(p, Q)/d \rceil$. ■

We now use Lemma 4.1 to bound $\mu(p)$ in terms of $\tau(p, P)$.

Theorem 4.2. *Let $\mathcal{P}$ be a set of n uncertain points in the uniform unipoint model, that is, each point is chosen with the same probability $\gamma > 0$. Let P be the set of sites in $\mathcal{P}$. There is a constant $c > 0$ such that for any point $p \in \mathbb{R}^d$ with $\tau(p, P) = t$, we have $(1-\gamma)^t \leq 1 - \mu(p) \leq d \exp\left(-\frac{\gamma t}{cd^2}\right)$.*

Proof: For the first inequality, fix a closed halfspace H that contains t points of P. If none of these t points is chosen then p does not appear in the convex hull of the outcome, so $1 - \mu(p) \geq (1-\gamma)^t$.

Next, let $\mathcal{S}$ be the set of simplices of Lemma 4.1, and let V be its set of vertices, where $T \geq \lceil t/d \rceil$. Let $n' = |V| = (d+1)T$. Set $\varepsilon = \frac{1}{d+1}$. A random subset of V of size $O(\frac{d}{\varepsilon} \log \frac{1}{\varepsilon\delta}) = O(d^2 \log \frac{d}{\delta})$ is an ε-net for halfspaces, with probability at least $1 - \delta$.

In particular, any halfspace passing through p, contains at least T points of V. That is, all these halfspaces are ε-heavy and would be stabbed by an ε-net. Now, if we pick each point of V with probability γ, it is not hard to argue that the resulting sample R is an ε-net[3]. Indeed, the expected size (and in with sufficiently large probability) of $R \cap V$ is $n'' = n'\gamma = (d+1)T\gamma \geq t\gamma$. As such, for some constant c, we need the minimal value of δ such that the inequality $t\gamma \geq cd^2 \ln \frac{d}{\delta}$ holds, which is equivalent to $\exp\left(\frac{t\gamma}{cd^2}\right) \geq \frac{d}{\delta}$. This in turn is equivalent to $\delta \geq d \exp\left(-\frac{t\gamma}{cd^2}\right)$. Thus, we set $\delta = d \exp\left(-\frac{t\gamma}{cd^2}\right)$.

Now, with probability at least $1 - \delta$, for a point p in $\mathbb{R}^d$, with Tukey depth at least t, we have that p is in the convex-hull of the sample. ■

Remark. Theorem 4.2 can be extended to the multipoint model. Assuming that each uncertain point has n_i sites and each site is chosen with probability γ, one can show that $(1-\gamma)^t \leq 1 - \mu(p) \leq d \exp\left(-\frac{\gamma t}{cd^2 n^*}\right)$, where $n^* = \max_{1 \leq i \leq m} n_i$.

Theorem 4.2 can be extended to the case when each point p_i of $\mathcal{P}$ is chosen with different probability, say, γ_i. In order to apply Theorem 4.2, we convert $\mathcal{P}$ to a multiset $\mathcal{Q}$, as follows. We choose a parameter $\eta = \frac{\delta}{10n}$. For each point $p_i \in \mathcal{P}$, we make $w_i = \left\lceil \dfrac{\ln(1-\gamma_i)}{\ln(1-\eta)} \right\rceil$ copies of p_i, each of which is selected with probability η. We can apply Theorem 4.2 to $\mathcal{Q}$ and show that if $\tau(q, \mathcal{Q}) \geq \frac{d^2}{\eta} \ln(2d/\delta)$, then $\mu(q, \mathcal{Q}) \geq (1 - \delta/2)$. Omitting the further details, we conclude the following.

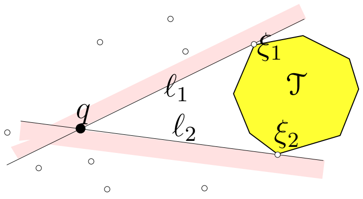
[2] If the points in Q are unweighted, then $\tau(q, Q)$ is simply the minimum number of points that lie in a closed halfspace that contains q.

[3] The standard argument uses slightly different sampling, but this is a minor technicality, and it is not hard to prove the ε-net theorem with this modified sampling model.

Corollary 4.3. *Let $\mathcal{P} = \{(p_1, \gamma_1), \ldots, (p_n, \gamma_n)\}$ be a set of n uncertain points in $\mathbb{R}^d$ under the unipoint model. For $1 \leq i \leq n$, set $w_i = \left\lceil \frac{\ln(1-\gamma_i)}{\ln(1-\delta/10n)} \right\rceil$ be the weight of point p_i. If the (weighted) Tukey depth of a point $q \in \mathbb{R}^d$ in $\{p_1, \ldots, p_n\}$ is at least $\frac{10d^2n}{\delta} \ln(2d/\delta)$, then $\mu(q, \mathcal{P}) \geq 1 - \delta$.*

Data structure. Let $\mathcal{P}$ be a set of points in the uniform unipoint model in $\mathbb{R}^2$, i.e., each point appears with probability γ. We now describe a data structure to estimate $\mu(q)$ for a query point $q \in \mathbb{R}^2$, within additive error $1/n$. We fix a parameter $t_0 = \frac{c}{\gamma} \ln n$ for some constant $c > 0$. Let $\mathcal{T} = \left\{x \in \mathbb{R}^2 \mid \tau(x, \mathcal{P}) \geq t_0\right\}$ be the set of all points whose Tukey depth in P is at least t_0. $\mathcal{T}$ is a convex polygon with $O(n)$ vertices [Mat91]. By Theorem 4.2, $\mu(q) \geq 1 - 1/n^2$ for all points $q \in \mathcal{T}$, provided that the constant c is chosen appropriately. We also preprocess P for halfspace range reporting queries [CGL85]. $\mathcal{T}$ can be computed in time $O(n \log^3 n)$ [Mat91], and constructing the half-plane range reporting data structure takes $O(n \log n)$ time [CGL85]. So the total preprocessing time is $O(n \log^3 n)$, and the size of the data structure is linear.

A query is answered as follows. Given a query point $q \in \mathbb{R}^2$, we first test in $O(\log n)$ time whether $q \in \mathcal{T}$. If the answer is yes, we simply return 1 as $\mu(q)$. If not, we compute in $O(\log n)$ time the two tangents ℓ_1, ℓ_2 of $\mathcal{T}$ from q. For $i = 1, 2$, let $\xi_i = \ell_i \cap \mathcal{T}$, and let ℓ_i^- be the half-plane bounded by ℓ_i that does not contain $\mathcal{T}$. Set $\mathcal{P}_q = \mathcal{P} \cap (\ell_1^- \cup \ell_2^-)$ and $n_q = |\mathcal{P}_q|$. Let R_q be the subset of $\mathcal{P}_q$ by choosing each point with probability γ.



By querying the half-plane range reporting data structure with each of these two tangent lines, we compute the set $\mathcal{P}_q$ in time $O(\log n + n_q)$. Let $\omega_q = \Pr[\, q \notin \text{CH}(R_q \cup \mathcal{T}) \,]$. We compute ω_q, in $(n_q \log n_q)$ time, by adapting the algorithm for computing $\mu(q)$ described in Section 2.

The correctness and efficiency of the algorithm follow from the following lemma, whose proof is omitted from this version.

Lemma 4.4. *For any point $q \notin \mathcal{T}$, (i) $|\Pr[\, q \in \text{CH}(R_q \cup \mathcal{T}) \,] - \mu(q)| \leq 1/n$; (ii) $n_q \leq 4t_0 = O(\gamma^{-1} \log n)$.*

By Lemma 4.4, $n_q = O(\gamma^{-1} \log n)$, so the query takes $O(\gamma^{-1} \log(n) \log \log n)$ time. We thus obtain the following.

Theorem 4.5. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$ in the unipoint model, where each point appears with probability γ. $\mathcal{P}$ can be preprocessed in $O(n \log^3 n)$ time into a linear-size data structure so that for any point $q \in \mathbb{R}^2$, returns a value $\widetilde{\mu}(q)$ in $O(\gamma^{-1} \log(n) \log \log n)$ time such that $|\widetilde{\mu}(q) - \mu(q) \,| \leq 1/n$.*

5. β-Hull

In this section, we consider the multipoint model, i.e., $\mathcal{P}$ is a set of m uncertain point defined by the pairs $\{(P_1, \Gamma_1), \ldots, (P_m, \Gamma_m)\}$. A convex set $C \subseteq \mathbb{R}^2$ is called *β-dense* with respect to $\mathcal{P}$ if it contains β-fraction of each (P_i, Γ_i), i.e., $\sum_{p_i^j} \gamma_i^j \geq \beta$ for all $i \leq m$. The *β-hull* of $\mathcal{P}$, denoted by $\text{CH}_\beta(\mathcal{P})$, is the intersection of all convex β-dense sets with respect to $\mathcal{P}$. Note that for $m = 1$, $\text{CH}_\beta(\mathcal{P})$ is the set of points whose Tukey depth is at least $1 - \beta$. We first prove an $O(n)$ upper bound on the complexity of $\text{CH}_\beta(\mathcal{P})$ and then describe an algorithm for computing it.

Theorem 5.1. *Let $\mathcal{P} = \{(P_1, \Gamma_1), \ldots, (P_m, \Gamma_m)\}$ be a set of m uncertain points in $\mathbb{R}^2$ under the multipoint model with $P = \bigcup_{i=1}^m P_i$ and $|P| = n$. For any $\beta \in [0, 1]$, $\text{CH}_\beta(\mathcal{P})$ has $O(n)$ vertices.*

Proof: We call a convex β-dense set C *minimal* if there is no convex β-dense set C' such that $C' \subset C$. A minimal convex β-dense set C is the convex hull of $P \cap C$. Therefore C is a convex polygon whose vertices are a subset of P. Obviously $\textsc{ch}_\beta(\mathcal{P})$ is the intersection of minimal convex β-dense sets. Therefore each edge of $\textsc{ch}_\beta(\mathcal{P})$ lies on a line passing through a pair of points of P, i.e.,$\textsc{ch}_\beta(\mathcal{P})$ is the intersection of a set H of halfplanes whose bounding line passes through a pair of points of P. Next we argue that $|H| \leq 2n$.

Fix a point $p \in P$. We claim that H contains at most two halfplanes whose bounding lines pass through p. Indeed if $p \in \text{int}(\textsc{ch}_\beta(\mathcal{P}))$, then no bounding line of H passes through p; if $p \in \partial(\textsc{ch}_\beta(\mathcal{P}))$, then at most two bounding lines of H pass through p; and if $p \notin \textsc{ch}_\beta(\mathcal{P})$, then there are two tangent to$\textsc{ch}_\beta(\mathcal{P})$ from p. Hence at most two bounding lines of H pass through p, as claimed. ■

We describe a property of the set of lines supporting the edges of $\textsc{ch}_\beta(\mathcal{P})$, which will be useful for computing $\textsc{ch}_\beta(\mathcal{P})$. We call a line ℓ passing through a point $p \in P_i$ *β-tangent* of P_i at p if one of the open half-planes bounded by ℓ contains less than β-fraction of points of P_i but the corresponding closed half-plane contains at least β-fraction of points. Using a simple perturbation argument, the following can be proved.

Lemma 5.2. *A line supporting an edge of $\textsc{ch}_\beta(\mathcal{P})$ is β-tangent at two points of P.*

Algorithm. We describe the algorithm for computing the upper boundary $\mathcal{U}$ of$\textsc{ch}_\beta(\mathcal{P})$. The lower boundary of $\textsc{ch}_\beta(\mathcal{P})$ can be computed analogously. It will be easier to compute $\mathcal{U}$ in the dual plane. Let $\mathcal{U}^*$ denote the dual of $\mathcal{U}$.

Recall that the dual of a point $p = (a, b)$ is the line $p^* : y = ax - b$, and the dual of a line $\ell : y = mx + c$ is the point $\ell^* = (m, -c)$. The point p lies above/below/on the line ℓ if and only if the dual point ℓ^* lies above/below/on the dual line p^*. Set $P_i^* = \left\{p_i^{j*} \mid p_i^j \in P_i\right\}$ and $P^* = \bigcup_{i=1}^m P_i^*$. For a point $q \in \mathbb{R}^2$ and for $i \leq m$, let $\kappa(q, i) = \sum \gamma_i^j$ where the summation is taken over all points $p_i^j \in P_i$ such that q lies below the dual line p_i^{j*}. We define the β-level Λ_i of P_i^* to be the upper boundary of the region $\left\{q \in \mathbb{R}^2 \mid \kappa(q, i) \geq \beta\right\}$. Λ_i is an x-monotone polygonal chain composed of the edges of the arrangement $\mathcal{A}(P_i^*)$. Further, the dual line of a point on Λ_i is a β-tangent line of P_i. Let Λ be the lower envelope of $\Lambda_1, \ldots, \Lambda_m$.

Let ℓ be the line supporting an edge of $\mathcal{U}$. Using Lemma 5.2, it can be argued that the dual point ℓ^* is a vertex of Λ. Next, let q be a vertex of $\mathcal{U}$, then q cannot lie above any β-tangent line of any P_i, which implies that the dual line q^* passes through a pair of vertices of Λ and does not lie below any vertex of Λ. Hence, each vertex of $\mathcal{U}$ corresponds to an edge of the upper boundary of the convex hull of Λ. By Theorem 5.1, $\mathcal{U}^*$, the dual of $\mathcal{U}$, has $O(n)$ vertices.

We now describe the algorithm for computing $\mathcal{U}^*$, which is similar to the one used for computing the convex hull of a level in an arrangement of lines [ASW08, Mat91]. We begin by describing a simpler procedure, which will be used as a subroutine in the overall algorithm.

Lemma 5.3. *Given a line ℓ, the intersection points of ℓ and Λ can be computed in $O(n \log n)$ time.*

Proof: We sort the intersections of the lines of P^* with ℓ. Let $\langle q_1, \ldots, q_u \rangle$, $u \leq n$, be the sequence of these intersection points. For every $i \leq m$, $\kappa(q_1, i)$ can be computed in a total of $O(n)$ time. Given $\{\kappa(q_{j-1}, i) \mid 1 \leq i \leq m\}$, $\{\kappa(q_j, i) \mid 1 \leq i \leq m\}$ can be computed in $O(1)$ time. A point $q_j \in \Lambda$ if $q_j \in \Lambda_i$ for some i and lies below Λ_i' for all other i'. This completes the proof of the lemma. ■

The following two procedures can be developed by plugging Lemma 5.3 into the parametric-search technique [ASW08, Mat91].

(A) Given a point q, determine whether q lies above $\mathcal{U}^*$ or return the tangent lines of $\mathcal{U}^*$ from q. This can be done in $O(n \log^2 n)$ time.

(B) Given a line ℓ, compute the edges of $\mathcal{U}^*$ that intersect ℓ, in $O(n \log^3 n)$ time. (Procedure (B) uses (A) and parametric search.)

Given (B), we can now compute $\mathcal{U}^*$ as follows. We fix a parameter $r > 1$ and compute a $(1/r)$-cutting[4] $\Xi = \{\Delta_1, \ldots, \Delta_u\}$, where $u = O(r^2)$. For each Δ_i, we do the following. Using (B) we compute the edges of $\mathcal{U}^*$ that intersect $\partial\Delta_i$. We can then deduce whether Δ_i contains any vertex of $\mathcal{U}^*$. If the answer is yes, we solve the problem recursively in Δ_i with the subset of lines of P^* that cross Δ_i. We omit the details from here and conclude the following.

Theorem 5.4. *Given a set $\mathcal{P}$ of uncertain points in $\mathbb{R}^2$ under the multipoint model with a total of n sites, and a parameter $\beta \in [0, 1]$, the β-hull of $\mathcal{P}$ can be computed in $O(n \log^3 n)$ time.*

Acknowledgments. P. Agarwal and W. Zhang are supported by NSF under grants CCF-09-40671, CCF-10-12254, and CCF-11-61359, by ARO grants W911NF-07-1-0376 and W911NF-08-1-0452, and by an ERDC contract W9132V-11-C-0003. S. Har-Peled is supported by NSF grants CCF-09-15984 and CCF-12-17462. S. Suri and H. Yıldız are supported by NSF grants CCF-1161495 and CNS-1035917.

References

[AAH$^+$13] P. K. Agarwal, B. Aronov, S. Har-Peled, J. M. Phillips, K. Yi, and W. Zhang. Nearest neighbor searching under uncertainty II. In *Proc. 32nd ACM Sympos. Principles Database Syst.*, pages 115–126, 2013.

[ACTY09] P. K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi. Indexing uncertain data. In *Proc. 28th ACM Sympos. Principles Database Syst.*, pages 137–146, 2009.

[ASW08] P.K. Agarwal, M. Sharir, and E. Welzl. Algorithms for center and tverberg points. *ACM Trans. Algo.*, 5(1):5:1–5:20, December 2008.

[CGL85] B. Chazelle, L. J. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, 25(1):76–90, 1985.

[Cha93] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(1):145–158, 1993.

[DRS09] N. N. Dalvi, C. Ré, and D. Suciu. Probabilistic databases: Diamonds in the dirt. *Commun. ACM*, 52(7):86–94, 2009.

[Eck93] J. Eckhoff. Helly, radon, and Carathéodory type theorems. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, pages 389–448. North-Holland, 1993.

[JLP11] A. Jørgensen, M. Löffler, and J. Phillips. Geometric computations on indecisive points. In *Proceedings of the 12th International Conference on Algorithms and Data Structures*, Proc. 12th Workshop Algorithms Data Struct., pages 536–547. Springer-Verlag, 2011.

[KCS11a] P. Kamousi, T. M. Chan, and S. Suri. Closest pair and the post office problem for stochastic points. In *Proc. 12th Workshop Algorithms Data Struct.*, pages 548–559, 2011.

[4] A $(1/r)$-*cutting* of P^* is a triangulation Ξ of $\mathbb{R}^2$ such that each triangle of Ξ crosses at most n/r lines of P^*.

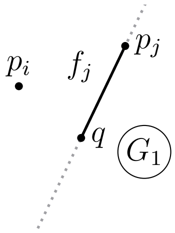


Figure 3:

[KCS11b] P. Kamousi, T.M. Chan, and S. Suri. Stochastic minimum spanning trees in euclidean spaces. In *Proc. 27th Annu. ACM Sympos. Comput. Geom.*, pages 65–74, 2011.

[Lö09] M. Löffler. *Data Imprecision in Computational Geometry.* PhD thesis, Dept. Computer Sci., 2009.

[Mat91] J. Matoušek. Computing the center of planar point sets. In J. E. Goodman, R. Pollack, and W. Steiger, editors, *Computational Geometry: Papers from the DIMACS Special Year*, pages 221–230. Amer. Math. Soc., 1991.

[MS92] J. Matoušek and O. Schwarzkopf. Linear optimization queries. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 16–25, 1992.

[Phi09] J.M. Phillips. *Small and Stable Descriptors of Distributions for Geometric Statistical Problems.* PhD thesis, Dept. Computer Sci., 2009.

[Sei04] R. Seidel. Convex hull computations. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 495–512. CRC Press, 2004.

[SVY13] S. Suri, K. Verbeek, and H. Yıldız. On the most likely convex hull of uncertain points. In *Proc. 21st Annu. European Sympos. Algorithms*, pages 791–802, 2013.

[ZYN12] Z. Zhao, D. Yan, and W. Ng. A probabilistic convex hull query tool. In *Proc. 15th Int. Conf. on Ext. Database Tech.*, pages 570–573, 2012.

A. Proof of Lemma 2.2

B. Computing Face Probabilities in Radial Order

Similar to the planar case, we can improve the computation time for the ith term to $O(n^{d-1})$ by considering the facets $f(P_\alpha)$ in radial order. In particular, let $L_\beta \subseteq P$ be a subset of $(d-2)$ sites. Let f_j denote the $(d-1)$-dimensional simplex $f(L_\beta \cup \{q\} \cup \{p_j\})$ where $p_j \notin L_\beta$ and $p_j \neq p_i$. We can compute the probability that f_j is a facet of C for all facets f_j in constant amortized time as follows. We project all sites to the two-dimensional plane passing through q and orthogonal to the $(d-2)$-dimensional hyperplane defined by $L_\beta \cup \{q\}$. (Such a plane is known as an orthogonal complement.) The hyperplane defined by $L_\beta \cup \{q\}$ projects onto q on this plane. Moreover, each facet f_j projects to a line segment extending from q. When we need to compute the probability that f_j is a facet of C, the set G_1 includes the sites on the other side of the line supporting f_j's projection with respect to p_i. (See Figure 3.) We compute probabilities for the facets f_j based on their radial order around q. The probability for the next facet in the sweep can be computed by modifying the probability of the previous facet in constant amortized time as we have done for the planar case, as we can efficiently track how G_1 changes. As a final note, we point out that the total cost of all sorting involved is $O(n^{d-1} \log n)$ which is less than the overall cost of $O(n^d)$.

C. Membership Probability Algorithms in the Multipoint Model

C.1. The Planar Case

Let $\mathcal{P}$ be an uncertain set of points in the multipoint model defined by site groups $\{P_1, \ldots, P_m\}$. We denote the jth site in p_i by p_i^j and its probability by γ_i^j. For simplicity, we set $n_i = |P_i|$. We define P to be the set all sites, i.e. $P = \bigcup_{1 \le i \le m} p_i$, and set $n = |P| = \sum_{1 \le i \le m} n_i$. Under this setting, we want to compute the membership probability of a given point q. Recall that the sites from a single site group p_i are dependent, i.e., they cannot co-exist in an outcome $B \subseteq P$ of the probabilistic experiment.

The algorithm for the unipoint model easily extends to the multipoint model. The main difference is the way we compute the probability for an edge. The rest of the algorithm remains the mostly the same. We now see this in more depth.

Let V and C be defined as before. As in the unipoint model, q is in the convex hull of B if and only if $q \in V$. We follow a similar strategy and decompose $\Pr[\, q \in V \,]$ as follows:

$$\Pr[\, q \in V \,] = \Pr[\, B = \emptyset \,] + \sum_{\substack{1 \le i \le m \\ 1 \le j \le n_i}} \Pr\big[\, qp_i^j \text{ is the witness edge of } q \notin \textsc{ch}(B) \,\big] \,.$$

The first term is trivial to compute in $O(n)$ time. We compute the probability that qp_i^j forms a witness edge of B as follows. Let $G_{i,j}$ be the set of sites to the right of the line $\overleftrightarrow{qp_i^j}$ where the right direction is with respect to the vector $\overrightarrow{qp_i^j}$. As in the unipoint model, the segment qp_i^j is the witness edge of B if and only if $p_i^j \in B$ and $B \cap G_{i,j} = \emptyset$. We can write the corresponding probability as follows:

$$\begin{aligned}
\Pr\big[\, p_i^j \in B \;\wedge\; B \cap G_{i,j} = \emptyset \,\big] &= \Pr\big[\, p_i^j \in B \,\big] \times \Pr\big[\, B \cap G_{i,j} = \emptyset \;\mid\; p_i^j \in B \,\big] \\
&= \Pr\big[\, p_i^j \in B \,\big] \times \prod_{1 \le k \le m} \Pr\big[\, B \cap G_{i,j} \cap P_k = \emptyset \;\mid\; p_i^j \in B \,\big] \\
&= \Pr\big[\, p_i^j \in B \,\big] \times \prod_{\substack{1 \le k \le m \\ k \ne i}} \Pr[\, B \cap P_k \cap G_{i,j} = \emptyset \,] \\
&= \gamma_i^j \times \prod_{\substack{1 \le k \le m \\ k \ne i}} \left(1 - \sum_{l \,|\, p_k^l \in G_{i,j}} \gamma_k^l \right) .
\end{aligned}$$

This expression can be easily computed in $O(n)$ time. It follows that one can compute $\overline{\mu(q)}$, thus $\mu(q)$, in $O(n^2)$ time.

As before, the computation time can be improved to $O(n \log n)$ by computing the witness edge probabilities in radial order around q. Let the circular sequence $p'_(1), p'_(2), \ldots, p'_(n)$ be the counterclockwise order of all sites around q, where each $p'_(i)$ is a distinct site p_a^b. We first compute the probability that $qp'_(1)$ is the witness edge in $O(n)$ time and also remember the values of the intermediate factors used in the computation. (The factors inside the $\prod_{1 \le k \le m}$ expression.) Then, for increasing values of i

from 2 to n, we compute the probability that $qp'_(k)$ is the witness edge by updating the probability for $qp'_(k-1)$. As a first step to this update, we update the values of the intermediate factors. To be more specific, let W_i denote the set of sites in the open wedge bounded by the lines $\overleftrightarrow{qp'_(i)}$ and $\overleftrightarrow{qp'_(i-1)}$. Also, for simplicity, assume that $p'_(k) = p_a^b$ and $p'_(k-1) = p_c^d$. Notice that $G_{a,b} = G_{c,d} \cup \left\{p_c^d\right\} \setminus W_i$. Then, for each site p_k^l in W_i, the kth factor increases by γ_k^l. Also, the cth factor decreases by γ_c^d. Finally, we temporarily set the value of the ath factor to 1 (to cover the case $k \neq i$ in the expression). Then, we can compute the witness edge probability for $qp'_(k)$ by multiplying the probability of $qp'_(k-1)$ with γ_a^b / γ_c^d and the multiplicative change in each intermediate factor. The cost of a single update is $O(1)$ amortized, as each site can contribute to at most 4 updates as in the unipoint case.

C.2. The d-dimensional case

All of the arguments in the d-dimensional algorithm are also easily extended to the multipoint. As before, we compute $\mu(q)$ by computing the probability $\Pr[\, q \in V\,]$. Following the same strategy, we decompose it as

$$\begin{aligned}\Pr\Big[\, q \in V \,\Big] &= \Pr\Big[\, q = \lambda(B \cup \{q\}) \,\Big] \\ &\quad + \sum_{1 \le i \le m} \left(\sum_{1 \le j \le n_i} \Pr\Big[\, p_i^j = \lambda(B \cup \{q\}) \ \wedge \ q \in V \,\Big] \right).\end{aligned}$$

It is trivial to compute the first term in $O(n)$ time. We now show how to compute each term inside the summations in $O(n^{d-1})$ time. This implies a total time of $O(n^d)$.

Clearly, Lemma 2.2 extends to the multipoint model, so we can use p_i^j-escaping facets to decompose our probability. Given a subset of sites$P_\alpha \subseteq P \setminus \left\{p_i^j\right\}$ of size $(d-1)$, define $f(P_\alpha)$ to be the $(d-1)$-dimensional simplex whose vertices are the points in P_α and q. Then,

$$\begin{aligned}\Pr\Big[\, p_i^j = \lambda(B \cup \{q\}) \ \wedge \ q \in V \,\Big] &= \Pr\Big[\, p_i^j = \lambda(B \cup \{q\}) \ \wedge \ q' \in V' \,\Big] \\ + \sum_{\substack{P_\alpha \subseteq P \setminus \left\{p_i^j\right\} \\ |P_\alpha| = (d-1) \\ f(P_\alpha) \text{ is } p_i^j\text{-escaping for } q}} &\Pr\Big[\, p_i^j = \lambda(B \cup \{q\}) \ \wedge \ f(P_\alpha) \text{ is a facet of } C \,\Big].\end{aligned}$$

The first term is computed recursively. We compute each term of the summation as follows. Let I_α be the set of group indices of the sites in P_α, i.e., $I_\alpha = \{u \mid \exists v \,.\, p_u^v \in P_\alpha\}$. As before, let $G_1 \subseteq P$ be the subset of sites which are on the other side of the hyperplane supporting $f(P_\alpha)$ with respect to p_i^j. Let $G_2 \subseteq P$ be the subset of sites that are below p_i^j along the x_d-axis. Following the same strategy, we write the desired probability as the probability that all points in P_α and p_i^j exist in B, and all points in $G_1 \cup G_2$ are absent from B. This probability is clearly zero, if any of the following conditions hold:

- $P_\alpha \cap G_2 \neq \emptyset$.
- p_i^j has a higher x_d-coordinate than q.
- P_α contains any two sites from the same uncertain point P_k.
- P_α contains any site from p_i.

Otherwise, we can write the probability as follows:

$$\begin{aligned}
&\Pr\big[\, p_i^j \in B \ \wedge\ P_\alpha \cap B = P_\alpha \ \wedge\ B \cap (G_1 \cup G_2) = \emptyset \,\big] \\
&\quad = \Pr\big[\, p_i^j \in B \,\big] \times \Pr\big[\, P_\alpha \cap B = P_\alpha \mid p_i^j \in B \,\big] \times \\
&\qquad\quad \Pr\big[\, B \cap (G_1 \cup G_2) = \emptyset \mid p_i^j \in B \ \wedge\ P_\alpha \cap B = P_\alpha \,\big] \\
&\quad = \Pr\big[\, p_i^j \in B \,\big] \times \Pr\Big[\, P_\alpha \cap B = P_\alpha \,\Big] \times \\
&\qquad\quad \Pr\big[\, B \cap (G_1 \cup G_2) = \emptyset \mid p_i^j \in B \ \wedge\ P_\alpha \cap B = P_\alpha \,\big] \\
&\quad = \Pr\big[\, p_i^j \in B \,\big] \times \Pr\Big[\, P_\alpha \cap B = P_\alpha \,\Big] \times \\
&\qquad\quad \prod_{\substack{1 \le u \le m \\ u \ne i \\ u \notin I_\alpha}} \Big(\Pr[\, P_u \cap B \cap (G_1 \cup G_2) = \emptyset \,] \Big) \\
&\quad = \gamma_i^j \times \prod_{u,v \,|\, p_u^v \in P_\alpha} \gamma_u^v \times \prod_{\substack{1 \le u \le m \\ u \ne i \\ u \notin I_\alpha}} \left(1 - \sum_{v \,|\, p_u^v \in (G_1 \cup G_2)} \gamma_u^v \right).
\end{aligned}$$

The expression takes linear time to compute and thus summation term can be computed in $O(n^d)$ time. Then, by induction, the computation of the term for the site p_i^j takes $O(n^d)$ time. As before, we can improve the computation time each term to $O(n^{d-1})$ by considering the facets $f(P_\alpha)$ in radial order. This implies a total complexity of $O(n^d)$ for the algorithm.

D. Computing the Probability Map in $O(n^4)$ Time

In this section, we describe how to compute the ***probability map*** for a given set of uncertain points on the plane in $O(n^4)$ time, rather than $O(n^5 \log n)$. For simplicity, we show how to compute the probability associated with each ***face*** (or cell) of the probability map. Our algorithm can easily be adapted to compute the probabilities of edges and vertices as well, by taking care of degeneracies. Also, we assume that the input is given in the unipoint model, however, we briefly explain how to extend the algorithm to the multipoint model.

The high level idea of our algorithm is as follows. Recall that the structure of the probability map is an arrangement of $O(n^2)$ lines, containing $O(n^4)$ faces. We first compute the membership probability of one of the faces, say F, in $O(n \log n)$ time. We then compute the membership probabilities of the faces neighboring F, in $O(1)$ time per each, by modifying the probability of F. By repeatedly applying the same idea of expanding into the neighbors, we can compute the probability of all faces in $O(n^4)$ time.

To complete our algorithm, we now describe how we can compute the probability of a face F' by using the already computed probability of one of its neighbors F. Without loss of generality, assume that F and F' are separated by a vertical line passing through the sites p_i and p_j and F is to the left of F'. Notice that the boundary separating F and F' is only a segment of the vertical line and does not contain p_i or p_j. Now imagine that a point q moves through this boundary, crossing from F to F'. It is easy to see that the change in the membership probability of q is due to the changes in witness edge probabilities of the segments qp_i and qp_j, as other sites are irrelevant. We now describe the change in the witness edge probability of qp_i. The probability of qp_j changes analogously. The change in the probability of qp_i happens differently for two cases:

1. p_i **is above** p_j**:** Then, p_j switches from the right side of the line $\overleftrightarrow{qp_i}$ to its left side (where right direction is with respect to the vector $\vec{qp_i}$). Consequently, the probability of qp_i changes by a factor of $\frac{1}{\overline{\gamma_j}}$.

2. p_i **is below** p_j**:** Then, p_j switches from the left side of the line $\overleftrightarrow{qp_i}$ to its right side. Consequently, the probability of qp_i changes by a factor of $\overline{\gamma_j}$.

The changes clearly require constant time operations, and thus the membership probability of F' can be computed in $O(1)$ time.

The extension of this technique to the multipoint model is straightforward. The only major difference is that we need to remember (similar to what is done in Appendix C) the intermediate factors when switching from face to face, as updating the witness edge probabilities requires updating these factors first. The total cost of an update remains $O(1)$ because each face switch updates one intermediate factor of two witness edge probabilities.