Star-Based Separators for Intersection Graphs of c-Colored Pseudo-Segments

Mark de Berg
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Bart M. P. Jansen
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Jeroen S. K. Lamme
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

The Planar Separator Theorem, which states that any planar graph $\mathcal{G}$ has a separator consisting of $O(\sqrt{n})$ nodes whose removal partitions $\mathcal{G}$ into components of size at most $\frac{2n}{3}$, is a widely used tool to obtain fast algorithms on planar graphs. Intersection graphs of disks, which generalize planar graphs, do not admit such separators. It has recently been shown that disk graphs *do* admit so-called clique-based separators that consist of $O(\sqrt{n})$ cliques. This result has been generalized to intersection graphs of various other types of disk-like objects. Unfortunately, segment intersection graphs do not admit small clique-based separators, because they can contain arbitrarily large bicliques. This is true even in the simple case of axis-aligned segments.

In this paper we therefore introduce *biclique-based separators* (and, in particular, *star-based separators*), which are separators consisting of a small number of bicliques (or stars). We prove that any c-oriented set of n segments in the plane, where c is a constant, admits a star-based separator consisting of $O(\sqrt{n})$ stars. In fact, our result is more general, as it applies to any set of n pseudo-segments that is partitioned into c subsets such that the pseudo-segments in the same subset are pairwise disjoint. We extend our result to intersection graphs of c-oriented polygons. These results immediately lead to an almost-exact distance oracle for such intersection graphs, which has $O(n\sqrt{n})$ storage and $O(\sqrt{n})$ query time, and that can report the hop-distance between any two query nodes in the intersection graph with an additive error of at most 2. This is the first distance oracle for such types of intersection graphs that has subquadratic storage and sublinear query time and that only has an additive error.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Computational geometry, intersection graphs, biclique-based separators, distance oracles

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.12

Related Version *Full Version*: https://arxiv.org/abs/2511.05371

Funding MdB and BJ are supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

1 Introduction

Background. The celebrated Planar Separator Theorem by Lipton and Tarjan [24] states that any planar graph admits a balanced separator of size $O(\sqrt{n})$. More precisely, for any planar graph $G = (V, E)$ with n nodes there exists a set $S \subseteq V$ of size $O(\sqrt{n})$ whose removal partitions G into components of at most $\frac{2n}{3}$ nodes each. This fundamental tool has been used to develop efficient algorithms for many classic problems on planar graphs.

Geometric intersection graphs – graphs whose nodes correspond to objects in the plane and that have an edge between two nodes iff the corresponding objects intersect – are a generalization of planar graphs that have received widespread attention in computational

geometry, graph theory, and parametrized complexity. (For an overview of work in the latter area, we refer the reader to the survey by Xue and Zehavi [28].) Unfortunately, geometric intersection graphs do not admit balanced separators of sublinear size, because they can contain arbitrarily large cliques. This led De Berg et al. [11] to introduce so-called *clique-based separators*: balanced separators that consist of a small number of disjoint (but potentially large) cliques instead of a small number of nodes. They proved that any disk graph – and more generally, any intersection graph of convex fat objects in the plane – admits a clique-based separator of size $O(\sqrt{n})$. Here the *size* of a clique-based separator is the number of cliques it consists of. They also showed how to use such clique-based separators to obtain sub-exponential algorithms for various classic graph problems, including Independent Set, Dominating Set, and Feedback Vertex Set. Recently, De Berg et al. [12] showed that intersection graphs of pseudo-disks, and intersection graphs of geodesic disks inside a simple polygon, admit balanced separators consisting of $O(n^{2/3})$ cliques, and Aronov et al. [3] proved that intersection graphs of geodesic disks in any well-behaved metric in the plane admit balanced separators consisting of $O(n^{3/4+\varepsilon})$ cliques.

One may wonder if *all* geometric intersection graphs have sublinear clique-based separators. Unfortunately the answer is no, even for intersection graphs of horizontal and vertical line segments. The problem is that such graphs can contain arbitrarily large bicliques, and $K_{n,n}$ does not admit a sublinear clique-based separator. We therefore introduce *biclique-based separators*, which are separators consisting of bicliques, and we show that any set of horizontal and vertical segments admits a balanced separator consisting of a small number of bicliques. In fact, our result is stronger (as it uses *star graphs* in the separator, and not just any biclique) and it applies to a much wider class of intersection graphs, as discussed next.

Our contribution. Let $\mathcal{G} = (V, E)$ be an undirected graph with n nodes. A collection $S = \{S_1, \ldots S_t\}$ of (not necessarily induced) disjoint subgraphs from G is called a balanced[1] *biclique-based separator* for G if it has the following properties:

- Each subgraph S_i is a biclique.
- The removal from $\mathcal{G}$ of all subgraphs S_i and their incident edges partitions $\mathcal{G}$ into connected components with at most $\frac{2n}{3}$ nodes each.

The *size* of a biclique-based separator is the number of bicliques it is comprised of. If each biclique $S_i \in S$ is a star, then we call S a *star-based separator*. Note that the subgraphs S_i in a biclique-based separator (or: in a star-based separator) need not be induced subgraphs of $\mathcal{G}$. This is necessary to be able to handle large cliques. To avoid confusion between our new separators that are comprised of bicliques and the traditional separators that are comprised of individual nodes, we will refer to the latter as *node-based separators*.

We denote the intersection graph induced by a set V of n objects in the plane by $\mathcal{G}^\times[V]$. Thus, the nodes in $\mathcal{G}^\times[V]$ are in one-to-one correspondence with the objects in V and there is an edge between two objects $u, v \in V$ iff u intersects v. In Section 2 we prove that a star-based separator of size $O(\sqrt{n})$ exists for the intersection graph of any set V of axis-parallel segments. (The bound on the size of the separator is tight in the worst case.) In fact, we will prove that a star-based separator of size $O(\sqrt{n})$ exists for any set V of pseudo-segments[2] that is partitioned into subsets $V_1, \ldots, V_c$ such that the pseudo-segments from each V_i are disjoint from each other. In other words, each V_i is an independent set in $\mathcal{G}^\times[V]$. We call

[1] In the sequel we will often omit the adjective *balanced* and simply speak of *separators*.

[2] A set V of curves in the plane is a set of pseudo-segments if any two curves in V are either disjoint or intersect in a single point that is a proper crossing (and not a tangency).

such a set V a *c-colored set* of pseudo-segments, and we call $V_1, \ldots, V_c$ its color classes; see Figure 1(left). Note that a set of axis-parallel segments such that no two segments of the same orientation overlap, is a 2-colored set of pseudo-segments. More generally, a c-oriented set of line segments such that no two segments from the same orientation intersect, is a c-colored set of pseudo-segments. Intersection graphs of c-oriented segments are often referred to as c-DIR graphs, and when segments of the same orientation are not allowed to intersect, they are referred to as PURE c-DIR graphs [20, 21]. We show that a star-based separator of size $O(\sqrt{n})$ for a c-DIR graph can be computed in $O(n \log n)$ time, if the segments which induce this graph are given.

In Section 3 we extend our result to the case where V is a set of n constant-complexity c-oriented simple polygons, that is, a set of polygons without holes such that the set of edges of all polygons is a c-oriented set. Note that two polygons can intersect without having their boundaries intersect, namely when one polygon is completely contained in the other polygon – this is the main difficulty we need to handle when extending our results to polygons. Finally, in Section 3 we also present a straightforward greedy algorithm that computes a star-based separator of size $O(n^{2/3} \log^{2/3} n)$ for any string graph.

Application to distance oracles. A *distance oracle* for a (potentially weighted) graph $\mathcal{G} = (V, E)$ is a data structure that can quickly report the distance between two query nodes $s, t \in V$. Such queries can trivially be answered in $O(1)$ time if we store the distance between any two nodes in a distance table, but this requires $\Omega(n^2)$ storage. The challenge is to design distance oracles that use subquadratic storage. Unfortunately, this is not possible in general: any distance oracle must use $\Omega(n^2)$ bits of storage in the worst case, irrespective of the query time [27]. This is even true for distance oracles that approximate distances to within a factor strictly less than 3. Thus, work on distance oracles concentrated on special graph classes and, in particular, on planar graphs. More than two decades of research culminated in an exact distance oracle for weighted planar graphs that uses $O(n^{1+o(1)})$ storage and has $O(\log^2 n)$ query time [9]. For the unweighted case – in other words, if we are interested in the *hop-distance* – there is a $(1+\varepsilon)$-approximate distance oracle with $O(1/\varepsilon^2)$ query time and $O(n/\varepsilon^2)$ storage [22]. See the survey by Sommer [26] and the paper by Charalampopoulos et al. [9] for overviews of the existing distance oracles for various graph classes.

For geometric intersection graphs, only few results are known. Gao and Zhang [15], and Chan and Skrepetos [6], provide $(1+\varepsilon)$-approximate distance oracles with $O(n \log n)$ storage and $O(1)$ query time for weighted unit-disk graphs. No exact distance oracles that use subquadratic storage and have sublinear query time are known, even for unweighted unit-disk graphs. Very recently, Chang, Gao, and Le [8] presented an *almost exact* distance-oracle for unit-disk graphs (and, more generally, for intersection graphs of similarly sized, convex, fat pseudodisks) that uses $O(n^{2-1/18})$ storage and that can report the distance between two query nodes, up to an additive error[3] of 2, in $O(1)$ time. Another recent result is by Aronov, De Berg, and Theocharous [3], who presented an almost exact distance oracle that uses clique-based separators. For intersection graphs of geodesic disks in the plane, their oracle uses $O(n^{7/4+\varepsilon})$ storage, has $O(n^{3/4+\varepsilon})$ query time, and can report the hop-distance between two query points up to an additive error of 1. For Euclidean disks, the storage and preprocessing would be $O(n\sqrt{n})$ and $O(\sqrt{n})$, respectively. As we explain later, their approach also works with biclique-based separators; the only difference is that the additive

[3] In the most recent arxiv version [7] the error has been reduced to 1.

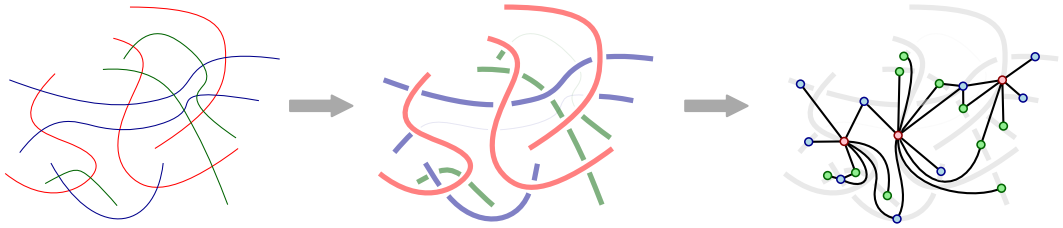

Figure 1 Left: A 3-colored set of pseudo-segments. Middle: The active fragments created by our algorithm. Right: The contact graph induced by the active fragments.

error increases from 1 to 2. Thus, we obtain an almost exact distance oracle for intersection graphs of c-colored pseudo-segments with $O(n\sqrt{n})$ storage and $O(\sqrt{n})$ query time. This is, to the best of our knowledge, the first almost exact distance oracle for intersection graphs of non-disk-like objects.

2 A star-based separator for c-colored pseudo-segments

Let V be a c-colored set of pseudo-segments, as defined above. To simplify the terminology, from now on we simply refer to the pseudo-segments in V as *segments*. We assume that the segments in V are in general position and, in particular, that no three segments meet in a common point and that no endpoint of one segment lies on another segment. This assumption is without loss of generality, as it can always be ensured by perturbing the segments slightly.

The construction. Recall that a *contact graph* is the intersection graph of a set of interior-disjoint objects. Contact graphs of curves are known to be planar if no four objects meet in a common point [19, Lemma 2.1]. Our strategy to construct a star-based separator S for $\mathcal{G}^{\times}[V]$ consists of the following steps, illustrated in Figure 1 and explained in more detail later.

Step 1. We partition each segment in V into *fragments*. Some fragments will be *active*, while others will be *inactive*. This partition will be such that active segments do not cross each other, although they may touch.

Step 2. Let $\mathcal{H}$ be the contact graph on the active segments. We construct a separator $S_{\mathcal{H}}$ on $\mathcal{H}$, using a suitable weighting scheme on the nodes of $\mathcal{H}$. Because $\mathcal{H}$ is planar, constructing the separator can be done using the Planar Separator Theorem.

Step 3. We use $S_{\mathcal{H}}$ to construct our star-based separator S for $\mathcal{G}^{\times}[V]$. For a fragment f, let $\text{seg}(f) \in V$ denote the segment containing f. Intuitively, we want to put a star into S for each fragment f in the separator $S_{\mathcal{H}}$, namely, the star consisting of the segment $\text{seg}(f)$ as well as all other segments intersecting $\text{seg}(f)$. For technical reasons, however, we actually have to put a slightly larger collection of stars into S.

To make this strategy work, we need to control the size of the contact graph $\mathcal{H}$. More precisely, to obtain a star-based separator of size $O(\sqrt{n})$, the size of $\mathcal{H}$ needs to be $O(n)$. Thus we cannot, for example, cut each segment into fragments at its intersection points with all other segments and make all the resulting fragments active. On the other hand, if we ignore certain parts of the segments by making them inactive, we miss certain intersections and we run the risk that our final set of stars is no longer a valid separator. Next, we describe how to overcome these problems by carefully creating the active fragments.

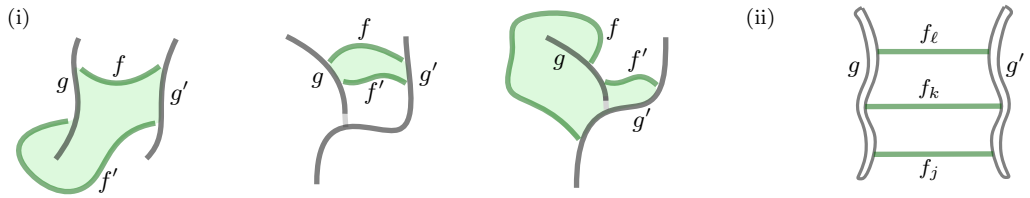

Figure 2 (i) Defining the region $Q(f, f', g, g')$. (ii) Illustration for the proof of Lemma 2.

Step 1: Creating the active fragments. To construct the active fragments that form the nodes in our contact graph $\mathcal{H}$, we will go over the subsets $V_1, \ldots, V_c$ one by one. We denote the active and inactive fragments created for a subset V_i by F_i and $\overline{F}_i$, respectively. For $1 \leqslant i \leqslant c$, we define $F_{\leqslant i} := F_1 \cup \cdots \cup F_i$, and we define $F_{<i}$ similarly.

Handling the first subset V_1 is easy: we simply define $F_1 := V_1$. In other words, each segment in V_1 becomes a single fragment, and all these fragments are active.

Now consider a subset V_i with $i > 1$. Each segment $v \in V_i$ is partitioned into one or more fragments by cutting it at every intersection point of v with an active fragment $f \in F_{<i}$. Let X_i be the set of fragments thus created. There are two types of fragments in X_i: fragments f that contain an endpoint of the segment $v \in V_i$ contributing f – there are at least one and at most two of these fragments per segment $v \in V_i$ – and fragments that do not contain such an endpoint. We call fragments of the former type *end fragments* and fragments of the latter type *internal fragments*. Note that an internal fragment has its endpoints on two distinct active fragments $g, g' \in F_{<i}$. We then say that f *connects* g and g'.

Now that we have defined X_i, we need to decide which fragments in X_i become active. To avoid making too many fragments active, we will partition X_i into equivalence classes, and we will activate only one fragment from each equivalence class. To define the equivalence classes we first define, for two internal fragments $f, f' \in X_i$ that connect the same pair of fragments $g, g' \in F_{<i}$, a region $Q(f, f', g, g')$, as follows; see Figure 2(i) for an illustration.

First, suppose that the segments g and g' do not touch each other, as in the left part of Figure 2(i). Thus, $\mathbb{R}^2 \setminus (g \cup g')$ is a single, unbounded region with two holes, namely g and g'. The fragment f connects these two holes, and so $\mathbb{R}^2 \setminus (g \cup g' \cup f)$ is still a single unbounded region, but now with one hole. Removing f' from this region splits it into two regions, one bounded and one unbounded. We define $Q(f, f', g, g')$ to be the bounded region. Now suppose that g and g' touch each other, say at an endpoint of g. We slightly shrink g at the point where it touches g', and then define $Q(f, f', g, g')$ as above. Note that in this case $Q(f, f', g, g')$ may consist of one or two bounded regions, if we undo the shrinking process; see the middle and right part in Figure 2(i). We can now define the equivalence classes.

▶ **Definition 1.** *Let f and f' be two fragments in X_i. We say that f and f' are* equivalent, *denoted by $f \equiv f'$, if $f = f'$ or the following two conditions hold.*

- **(i)** *The fragments f and f' are internal and connect the same pair of fragments $g, g' \in F_{<i}$.*
- **(ii)** *The region $Q(f, f', g, g')$ enclosed by the fragments f, f', g, g' does not contain an endpoint of any segment in V.*

The following lemma shows that $\equiv$ is indeed an equivalence relation.

▶ **Lemma 2.** *The relation $\equiv$ defined in Definition 1 is an equivalence relation.*

Proof. It is clear that $\equiv$ is reflexive and symmetric. It remains to show that if $f_1 \equiv f_2$ and $f_2 \equiv f_3$, then $f_1 \equiv f_3$. Let g, g' be the fragments connected by f_1, f_2, f_3. We can assume that g and g' do not touch; otherwise, as before, we can shrink one of the fragments so that

the arguments below still apply. It is instructive to view g and g' as being slightly inflated so that they become closed curves and no longer have two "sides". We then see that, from a topological point of view, the situation is always as in Figure 2(ii): two of the fragments, f_j and f_ℓ, are incident to the unbounded face, while the third fragment f_k is not. Thus, if we define $Q_{jk} := Q(f_j, f_k, g, g')$ and we define $Q_{k\ell}$ and $Q_{j\ell}$ similarly, then $Q_{j\ell} = Q_{jk} \cup Q_{k\ell}$. This implies that, no matter which of the three fragments f_j, f_k, f_ℓ is f_2, we always have $Q_{13} \subseteq Q_{12} \cup Q_{23}$. Since Q_{12} and Q_{23} do not contain endpoints of segments in V if $f_1 \equiv f_2$ and $f_2 \equiv f_3$, neither does Q_{13}. Thus, $f_1 \equiv f_3$. ◀

We now partition X_i into equivalence classes according to the relation $\equiv$ defined above. For each equivalence class, we make an arbitrary fragment $f \in X_i$ from that class active and put it into F_i; the other fragments from that equivalence class are made inactive and put into $\overline{F}_i$. Note that end fragments are always active, since they do not connect a pair of fragments from $F_{<i}$ and thus cannot be equivalent to any other fragment.

Step 2: Creating the contact graph $\mathcal{H}$ and its separator $S_\mathcal{H}$. Let $F := F_1 \cup \cdots \cup F_c$ be the set of active fragments created in Step 1, and let $\overline{F} := \overline{F}_1 \cup \cdots \cup \overline{F}_c$ be the inactive fragments. We define $\mathcal{H} = (F, E_\mathcal{H})$ to be the contact graph of F. More precisely, for two fragments $f, f \in F$ we put the edge (f, f') in $E_\mathcal{H}$ iff f and f' are in contact – that is, $f \cap f' \neq \emptyset$ – and they do not belong[4] to the same segment in V; see Figure 1. Because of our general position assumption, no three fragments from different segments meet in a point, and therefore $\mathcal{H}$ is planar [19, Lemma 2.1]. From now on, with a slight abuse of notation, we will not make a distinction between the nodes in $\mathcal{H}$ and the corresponding fragments in F.

We now wish to create a separator $S_\mathcal{H}$ for $\mathcal{H}$. In Step 3 we will use $S_\mathcal{H}$ to create a separator S for $\mathcal{G}^\times[V]$. To ensure that S will be balanced, we will put weights on the nodes in $\mathcal{H}$ and use a weighted version of the Planar Separator Theorem, as described next.

For each segment $s \in V$, we designate one of its end fragments – recall that end fragments are always active – as its *representative fragment*. We give all representative fragments a weight of $\frac{1}{n}$, and all other fragments a weight of 0. Note that the total weight of the fragments in F is 1. We apply the weighted separator theorem given below to $\mathcal{H}$. This gives us a separator $S_\mathcal{H}$, and parts $A_\mathcal{H}, B_\mathcal{H} \subseteq F \setminus S_\mathcal{H}$ such that there are no edges between $A_\mathcal{H}$ and $B_\mathcal{H}$.

▶ **Lemma 3** (Theorem 4 in [24]). *Let G be a non-negatively weighted planar graph containing n nodes whose weights sum up to at most 1. The node set V_G can be partitioned into a separator S, and sets A and B such that $|S| = O(\sqrt{n})$, no edge connects A and B, and the total weight of the nodes in A, as well as the total weight of the nodes in B, is at most $\frac{2}{3}$.*

Step 3: Creating the star-based separator S for $\mathcal{G}^\times[V]$. Using the separator $S_\mathcal{H}$ created in Step 2, we now create our star-based separator S for the intersection graph $\mathcal{G}^\times[V]$. We do this by putting one or three stars into S for each fragment $f \in S_\mathcal{H}$, as follows. For a segment $s \in V$, define $\text{star}(s)$ to be the subgraph of $\mathcal{G}^\times[V]$ consisting of s and all its incident edges. Thus, the nodes in $\text{star}(s)$ are the segment s itself plus the segments $s' \in V$ that intersect s.

- If $f \in S_\mathcal{H}$ is an end fragment then we put $\text{star}(\text{seg}(f))$ into S.
- If $f \in S_\mathcal{H}$ is an internal fragment then let $g, g' \in F$ be the pair of active fragments connected by f. We put $\text{star}(\text{seg}(f))$, $\text{star}(\text{seg}(g))$, and $\text{star}(\text{seg}(g'))$ into S.

[4] Since we do not put an edge between fragments belonging to the same segment, even if these fragments touch, $\mathcal{H}$ is formally speaking not a contact graph, but we permit ourselves this abuse of terminology.

Note that multiple copies of the same star can be added to S. We remove these duplicates to ensure that all star graphs in S have unique centers. It can still be the case that several stars in S contain the same node. To make the stars pairwise disjoint, we therefore remove non-center nodes until every node appears in at most one star in S.

The analysis. We now show that the construction described above yields a balanced separator of the required size. This requires proving two things: the *separation property*, namely that the removal of S partitions $\mathcal{G}^\times[V]$ into components of size at most $\frac{2n}{3}$, and the *size property*, namely that S consists of $O(\sqrt{n})$ stars.

Proving the separation property. To prove the separation property, it suffices to show that $V \setminus S$ can be partitioned[5] into subsets A and B such that $|A| \leqslant \frac{2n}{3}$ and $|B| \leqslant \frac{2n}{3}$, and such that no segment in A intersects any segment in B.

We define the sets A and B as follows. For each segment $v \in V$ not contained in a star in S we look at its representative fragment f_v. Note that f_v must be contained in either $A_\mathcal{H}$ or $B_\mathcal{H}$, since $f_v \in S_\mathcal{H}$ would imply that v is contained in a star in S. If $f_v \in A_\mathcal{H}$ then we add v to A, else we add v to B. The value $\frac{|A|}{n}$ can be at most the total weight of fragments in $A_\mathcal{H}$, and a similar statement holds for B. Hence, the next observation follows from the fact that $S_\mathcal{H}$ is a balanced separator.

▶ **Observation 4.** *$|A| \leqslant \frac{2n}{3}$ and $|B| \leqslant \frac{2n}{3}$.*

The more challenging part is to show that no segment in A intersects any segment in B. We will need the following lemma. Recall that the segment set V is partitioned into color classes $V_1, \ldots, V_c$, which we handled one by one to create the set F of active fragments.

▶ **Lemma 5.** *Let $v \in V_i \setminus S$ and let f be an active fragment. Suppose one of the following conditions holds:*

(i) *$f \in F_{<i}$ and v intersects f,*
(ii) *$\mathrm{seg}(f) = v$, or*
(iii) *f is equivalent to an inactive fragment f' such that $\mathrm{seg}(f') = v$.*

Then $f \in A_\mathcal{H}$ if $v \in A$, and $f \in B_\mathcal{H}$ if $v \in B$.

Proof. We prove the lemma under the assumption that $v \in A$; the proof for $v \in B$ is analogous. Let f_v be the representative fragment of v. Because $v \in A$, we have $f_v \in A_\mathcal{H}$. We will define sets Z_1, Z_2, Z_3 that contain the active fragments for which conditions (i), (ii), and (iii) hold, respectively, and then argue that the lemma holds for each of the three sets.

Let f'_v be the other end fragment of v, and let $Z_1 = \{g_1, \ldots, g_k\}$ be the ordered set of fragments from $F_{<i}$ that we cross as we trace v from f_v to f'_v; see Figure 3. It is possible that f'_v does not exist, in this case Z_1 is empty. Note that every pair $g_j, g_{j+1} \in Z_1$ is connected by an active or inactive fragment of v, which we denote by f_j. We now define Z_2, Z_3 as follows.

- Z_2 contains the fragments f_j that are active plus the end fragments f_v and (if it exists) f'_v. Thus, Z_2 simply contains all active fragments of v.
- Z_3 contains, for each inactive fragment f_j, the unique equivalent active fragment $f'_j \in F_i$.

It is easily checked that the sets Z_1, Z_2, Z_3 indeed contain exactly those fragments for which conditions (i), (ii), and (iii) hold, respectively. We will now prove that all fragments in $Z_1 \cup Z_2 \cup Z_3$ are in $A_\mathcal{H}$. To this end, observe that the fragments in $Z_1 \cup Z_2 \cup Z_3$ correspond

[5] Formally, we should have written $V \setminus \bigcup S$ instead of $V \setminus S$, since $S_\mathcal{H}$ is a set of stars and not a set of nodes, but we prefer the simpler (though technically incorrect) notation.

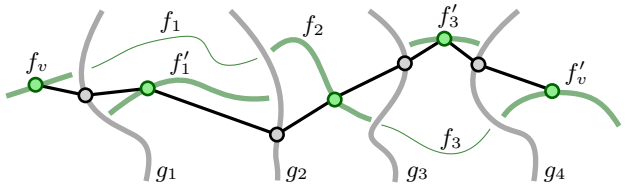

Figure 3 The various fragments used in the proof of Lemma 5. Note that the segments $\text{seg}(f'_1)$ and $\text{seg}(f'_3)$ are not drawn in their entirety. The path in $\mathcal{H}$ from f_v to f'_v is also shown.

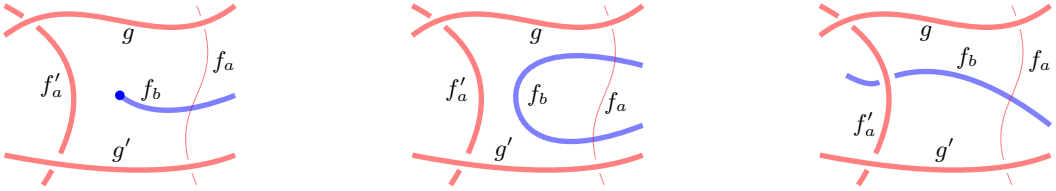

Figure 4 Left: The segment b has an endpoint in Q. Middle: The segment b intersects a twice. Right: The segment b intersects f'_a. All of these cases lead to a contradiction.

to nodes in $\mathcal{H}$ that form a path π starting at f_v and ending at f'_v, as illustrated in Figure 3. Recall that $f_v \in A_{\mathcal{H}}$. Now assume for a contradiction that there is a fragment $f \in Z_1 \cup Z_2 \cup Z_3$ that is in $B_{\mathcal{H}} \cup S_{\mathcal{H}}$. Because there are no edges between $A_{\mathcal{H}}$ and $B_{\mathcal{H}}$, this means there must be a fragment f^* on the subpath of π from f_v to f that is an element of the separator $S_{\mathcal{H}}$. If $f^* \in Z_1 \cup Z_2$ then $v \in \text{star}(\text{seg}(f^*))$, which contradicts that $v \in A$. Otherwise $f^* \in Z_3$ and f^* connects two fragments $g_j, g_{j+1} \in Z_1$. Since $f^* \in S_{\mathcal{H}}$, this implies that $\text{star}(\text{seg}(g_j))$ is a star in S. Because $v \in \text{star}(\text{seg}(g_j))$, this again contradicts that $v \in A$. ◀

We can now prove that $\mathcal{G}^\times[V]$ does not contain edges between nodes in A and nodes in B.

▶ **Lemma 6.** *No segment $a \in A$ intersects any segment $b \in B$.*

Proof. Assume for a contradiction that $a \in A$ and $b \in B$ intersect. Let $f_a \subseteq a$ and $f_b \subseteq b$ be the fragments containing the intersection point. We distinguish three cases.

- *Both f_a and f_b are active.* Then f_a and f_b satisfy condition (ii) of Lemma 5, and so $f_a \in A_{\mathcal{H}}$ and $f_b \in B_{\mathcal{H}}$. Because f_a and f_b are active and intersect, (f_a, f_b) is an edge in $\mathcal{H}$. But then $S_{\mathcal{H}}$ would not be a proper separator, and we reach a contradiction.
- *Both f_a and f_b are inactive.* Let $f_a \in \overline{F}_i$ and $f_b \in \overline{F}_j$, and assume without loss of generality that $i \leqslant j$. Because a and b intersect and segments from the same color class are disjoint, we cannot have $i = j$. Hence, $i < j$. Because f_a is inactive, it must be an internal fragment that connects some fragments $g, g' \in F_{<i} \subseteq F_{<j}$. Thus, g, g' satisfy condition (i) of Lemma 5, with a playing the role of v, which implies that $g, g' \in A_{\mathcal{H}}$. Let $f'_a \in F_i$ be the active fragment that is equivalent to f_a. Then f'_a satisfies condition (iii) from Lemma 5 and so $f'_a \in A_{\mathcal{H}}$. The fragments f_a, f'_a, g, g' enclose some region Q. The segment b intersects f_a so it is partly contained in Q; see Figure 4 for an illustration of the possibilities for f_b entering Q. It cannot have an endpoint in Q, because f_a and f'_a are equivalent. Segment b cannot intersect f_a twice, because V is a collection of pseudo-segments. It follows that b must intersect f_a, g or g'. Now observe that f_a, g and g' are all fragments in $F_{<j}$. This implies that the fragment that intersects b satisfies condition (i) from Lemma 5, with b playing the role of v. But then the intersected fragment would be in $B_{\mathcal{H}}$, contradicting that $A_{\mathcal{H}}$ and $B_{\mathcal{H}}$ are disjoint.

- *One of the fragments f_a, f_b is active and one is inactive.* Assume wlog that f_a is inactive and f_b is active. Let $f_a \in \overline{F}_i$ and $f_b \in F_j$. Because a and b intersect, we have $i \neq j$. If $i < j$ then the arguments from the previous case can be used to obtain a contradiction – indeed, these arguments did not use that f_b is inactive. If $i > j$ then $f_b \in F_{<i}$. Also note that a intersects f_b. But then f_b satisfies condition (i) from Lemma 5, with a playing the role of v. It follows that $f_b \in A_{\mathcal{H}}$, which contradicts that $A_{\mathcal{H}}$ and $B_{\mathcal{H}}$ are disjoint.

We have reached a contradiction in each case, thus proving the lemma. ◀

Proving the size property. Because we add at most three stars to S per fragment in $S_{\mathcal{H}}$, it suffices to bound the size of $S_{\mathcal{H}}$ to prove the size property. From Lemma 3 it follows that $|S_{\mathcal{H}}| = O(\sqrt{|F|})$. The next lemma bounds $|F|$.

▶ **Lemma 7.** *$|F| = O(n \cdot 4^c)$, where c is the number of color classes in V.*

Proof. We first bound $|F_i|$, the number of active fragments created for the segments in V_i, in terms of the number of active fragments created for $V_{<i}$.

▷ Claim. For $1 < i \leqslant c$ we have $|F_i| \leqslant 2 \cdot |V_i| + 3 \cdot (|F_{<i}| + 2n + 1) - 6$.

Proof. The set F_i contains at most $2 \cdot |V_i|$ end fragments. The internal fragments in F_i connect two fragments from $F_{<i}$. We denote the set of these internal fragments by F_i^{int}. Now consider the multi-graph $\mathcal{G}$ defined as follows.

- For each fragment in $F_{<i}$, we add a node to $\mathcal{G}$.
- For each fragment in F_i^{int}, we add an edge to $\mathcal{G}$ between the fragments from $F_{<i}$ that it connects.
- For each endpoint of a segment in $V_{\geqslant i}$ we add a singleton node to $\mathcal{G}$. (Observe that the endpoints of segments in $V_{<i}$ lie on an end fragment in $F_{<i}$, which is already a node in $\mathcal{G}$.)

Now consider the obvious drawing of $\mathcal{G}$, where the nodes are drawn as fragments of $F_{<i}$ or as points, and the edges are drawn as fragment in F_i^{int}. Recall that two active fragments can touch, but they never cross. By continuously shrinking the fragments in $F_{\leqslant i}$ and deforming the fragments of F_i^{int} appropriately, we can therefore create a plane drawing of $\mathcal{G}$. That is, we can create a drawing of $\mathcal{G}$ in which the nodes are points and the edges are pairwise disjoint curves connecting their endpoints. See Figure 5 for an example of this deformation. For reasons that will become clear shortly, we augment $\mathcal{G}$ with one additional singleton node u_∞, which we place in the unbounded face of $\mathcal{G}$.

The graph $\mathcal{G}$ is a multi-graph because F_i^{int} can contain multiple fragments connecting the same pair of fragments $f, f' \in F_{<i}$. Let $g, g' \in F_i^{\text{int}}$ be two such fragments. The reason that we added both g and g' to F_i^{int} is that g and g' were not equivalent. Hence, the region $Q(f, f', g, g')$ contains an endpoint belonging to some segment $v \in V$. The deformation process that turns each node the drawing of $\mathcal{G}$ into a point can be done in such a way that this property is maintained. Thus, after the deformation we have a plane drawing of $\mathcal{G}$ in which for any two edges g, g' that connect the same pair of nodes, there is a node inside the deformed region $Q(f, f', g, g')$. Because of the additional node u_∞, we are also guaranteed to have at least one node outside this region. A plane multi-graph with this property is called a *thin graph*. It is known [1, Lemma 5] that the standard inequality $(\#\text{edges}) \leqslant 3 \cdot (\#\text{nodes}) - 6$ that holds for planar graphs (with at least three vertices) also holds for thin graphs. Hence, $|F_i^{\text{int}}| \leqslant 3(|F_{<i}| + 2n + 1) - 6$. ◁

Note that $|F_1| = |V_1|$ and $|V_i| \leqslant n$ for all i. Hence, the claim above gives us the recurrence $|F_{\leqslant i}| \leqslant 4|F_{<i}| + 8n - 3$ with $|F_{\leqslant 1}| \leqslant n$. This gives $|F_{\leqslant i}| \leqslant \left(\frac{n}{4} + \frac{8n-3}{12}\right) \cdot 4^i - \frac{8n-3}{3}$. Plugging in $i = c$ gives $|F| = |F_{\leqslant c}| = O(n \cdot 4^c)$, which proves the lemma. ◀

Figure 5 Left: Fragments in V_i are green, fragments in F_i^{int} are dark gray, and endpoints of segments in $V_{\geqslant i}$ are blue. Right: The plane multigraph created for the example on the left.

Since c, the number of color classes, is a constant, we obtain the following corollary.

▶ **Corollary 8.** *The separator S contains $O(\sqrt{n})$ star graphs.*

So far we have considered input sets V where the segments are unweighted. To create a separator for c-oriented polygons, which we will do in the next section, we need a separator for weighted segments. Fortunately it is straightforward to adapt the construction described above to the weighted setting – we only need to change the weighting scheme we used in Step 2 of the construction. More precisely, instead of assigning a weight $\frac{1}{n}$ to the representative fragment of a segment v, we assign $\text{weight}(v)/\sum_{u\in V}\text{weight}(u)$ to the representative.

▶ Remark. It is well known that grid graphs do not admit node-based separators of size $o(\sqrt{n})$. Because nodes in grid graphs have constant degree, bicliques in grid graphs have constant size. Hence, grid graphs do not admit biclique-based separators of size $o(\sqrt{n})$. Grid graphs are bipartite and planar, which implies that they are PURE-2-DIR graphs [18]. We conclude that even PURE-2-DIR graphs do not admit biclique based separators of size $o(\sqrt{n})$.

Computation time. If we assume that the appropriate elementary operations on the pseudo-segments – computing the intersection point of two pseudo-segments, for instance, or determining if a point lies inside some region $Q(f, f', g, g')$ – can be performed in $O(1)$ time, that then a brute-force implementation of the algorithm presented above runs in polynomial time. More interestingly, for c-oriented line segments, the algorithm can be implemented to run in $O(n \log n)$ time, as shown in the full version of this paper. We obtain the following theorem.

▶ **Theorem 9.** *Let V be a c-colored set of n non-negatively weighted pseudo-segments, where c is a fixed constant, whose total weight is at most 1. Then the intersection graph $\mathcal{G}^\times[V]$ has a star-based separator S of size $O(\sqrt{n})$ such that $V \setminus S$ can be partitioned into subsets A, B of weight at most $\frac{2}{3}$ with no edges between them. The bound on the size of the separator is tight, even for axis-parallel segments. In the special case where V is a set of c-oriented line segments, the separator S and parts A, B can be computed in $O(n \log n)$ time.*

Application to distance oracles. Arikati et al. [2] presented a simple distance oracle for planar graphs, using node-based separators. Aronov, De Berg, and Theocharous [3] observed that the approach can be adapted to work with clique-based separators, as follows. Let $\mathcal{G} = (V, E)$ be the graph for which we want to construct a distance oracle.

- Construct a clique-based separator S for $\mathcal{G}$, and let $A, B \subseteq V \setminus S$ be the two parts of the partition given by S. For each node $v \in V$ and each clique $C \in S$, store the distance $d(v, C) := \min\{d(v, u) : u \in C\}$, where $d(u, v)$ denotes the hop-distance from s to t in $\mathcal{G}$.
- Recursively construct distance oracles for the subgraphs induced by the parts A and B.

Now suppose we want to answer a distance query with nodes $s, t \in V$. Let $d^* := \min\{d(s, C) + d(t, C) : C \in S\}$. If s and t do not lie in the same part – that is, we do not have $s, t \in A$ or $s, t \in B$ – then we report d^*. Otherwise, s and t lie in the same part of the partition, say A. Then we report the minimum of d^* and the distance we obtain by querying the recursively constructed oracle for A.

This distance oracle uses $O(n \cdot s(n))$ storage, where $s(n)$ is the size of the separator, and it has $O(s(n))$ query time, assuming $s(n) = \Omega(n^\beta)$ for some constant $\beta > 0$. The reported distance is either the exact distance $d(s, t)$, or it is $d(s, t) - 1$. The additive error of 1 is because we do not know if s and t can reach the same node of some clique C with paths of length $d(s, C)$ and $d(t, C)$, respectively – we may have to use an edge inside C to connect these paths. We observe that the same approach can be used in combination with star-based (or biclique-based) separators. The only difference is that we now get an additive error of at most 2, because we may need two additional edges inside a star (or biclique) in the separator. We obtain the following result.

▶ **Corollary 10.** *Let V be a c-colored set of pseudo-segments, where c is a fixed constant. There is an almost-exact distance oracle for $\mathcal{G}^\times[V]$ that uses $O(n\sqrt{n})$ storage and can report the hop-distance between any two nodes $s, t \in V$, up to an additive error of 2, in $O(\sqrt{n})$ time.*

3 Extension to c-oriented polygons and string graphs

c-Oriented polygons. Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a collection of c-oriented simple polygons, each with a constant number of edges, where c is a fixed constant. The idea is to create a weighted collection V of segments to which we can apply Theorem 9, and then use the resulting separator S_V to construct a separator S for $\mathcal{G}^\times[\mathcal{P}]$. The set V is created as follows.

- First, we add each side of every polygon $P_i \in \mathcal{P}$ to V. For each polygon P_i, we pick an arbitrary side as its *representative side*, which we give weight $\frac{1}{n}$; other sides of P_i are given weight 0.
- Second, to handle the containment of polygons within other polygons, we add so-called *containment segments* to V. These segments will always have weight 0. Let ϕ be an orientation that is not used by any segment in V. For each polygon P_i let $\mathcal{C}_i \subseteq \mathcal{P}$ be the set of polygons that fully contain P_i. Take a point $x_i \in P_i$, and let ρ_i be a ray emanating from x_i and with orientation ϕ. For each $P_j \in \mathcal{C}_i$, let y_j be the point where ρ_i leaves P_j for the first time. Define x_i' to be the last such point y_{j^*}. We now add $x_i x_i'$ as a containment segment to V. Note that the containment segment $x_i x_i'$ intersects the boundaries of all polygons in $\mathcal{C}_i$. Moreover, $x_i x_i'$ is completely contained in P_{j^*}. Figure 6 shows examples of containment segments.

Clearly, we can partition V into color classes $V_1, \ldots, V_{c+1}$ based on the orientation of the segments. We let V_1 contain the segments of orientation ϕ, which will be important later. The total weight of the segments is 1, so we can apply Theorem 9. Let S_V be the resulting separator for $\mathcal{G}^\times[V]$, and let A_V and B_V be the two parts of weight at most $\frac{2}{3}$ into which S_V splits $V \setminus S_V$. We construct a separator S and parts A, B for $\mathcal{G}^\times[\mathcal{P}]$ as follows.

For each star in S_V we consider its center v. If v is a side of a polygon $P_i \in \mathcal{P}$ then we add $\text{star}(P_i)$ to S, where $\text{star}(P_i)$ is the subgraph of $\mathcal{G}^\times[\mathcal{P}]$ consisting of P_i and its incident edges. If v is a containment segment $x_i x_i'$ that we generated for polygon $P_i \in \mathcal{P}$, then let P_j be the polygon enclosing P_i such that $x_i' \in \partial P_j$. We add $\text{star}(P_j)$ to S_P. As before, we remove duplicate stars and we remove polygons from stars to ensure that each polygon is in at most one star. To create the parts A, B, we consider the representative sides of the polygons $P_i \in \mathcal{P}$ that are not in a star in S. If the representative side is in A_V, then we put P_i in part A; otherwise we put P_i in part B.

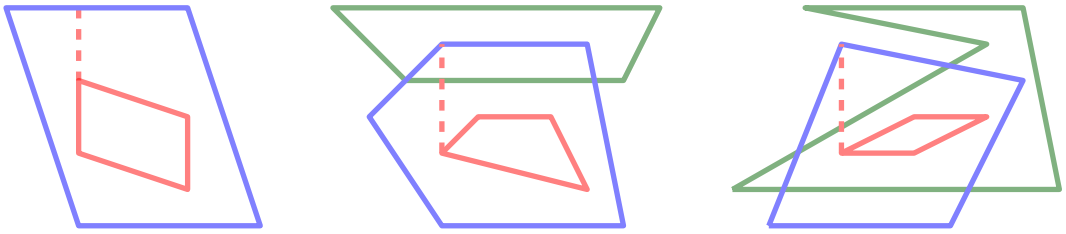

Figure 6 Left: The dashed line is the containment segment for the red polygon. Middle: The containment segment for the red polygon intersects a polygon which does not contain the red polygon, this is allowed. Right: The containment segment stops after intersecting the boundary of every polygon which contains the red polygon. Extending the containment segment could lead to further intersections with the green polygon. However, only the first intersection between the containment segment and a polygon are considered in the construction.

The analysis. Proving the size property, namely that $|S| = O(\sqrt{n})$, is easy. Indeed, we put $O(1)$ sides and one containment segment per polygon into V. Hence, $|V| = O(n)$ and thus $|S| = |S_V| = O(\sqrt{n})$. It remains to prove the separation property.

Since the total weight of A_V and of B_V are both at most $\frac{2}{3}$ and each representative side has weight $\frac{1}{n}$, it follows that $|A| \leqslant \frac{2n}{3}$ and $|B| \leqslant \frac{2n}{3}$. Next, we prove that there are no edges from A to B in $\mathcal{G}^{\times}[\mathcal{P}]$. We need the following lemma.

▶ **Lemma 11.** *Consider a polygon $P_i \in \mathcal{P}$. If $P_i \in A$ (resp. $P_i \in B$), then all sides of P_i are in A_V (resp. B_V).*

Proof. First consider the case $P_i \in A$. Suppose for a contradiction that not all sides of P_i are in A_V. Hence, P_i has a side $s \in S_V$ or a side $s' \in B_V$. We claim that in the latter case P_i must also have a side $s \in S_V$. To see this, consider the representative side s_i of P_i. Note that $s_i \in A_V$ since we put P_i into A. Also note that there is a path in $\mathcal{G}^{\times}[V]$ connecting s' to s_i – the nodes on this path correspond to the sides of P_i that we encounter as we traverse ∂P_i from s' to s_i in, say, clockwise order. Since $s_i \in A_V$ and $s' \in B_V$, one of the nodes on this path must be in S_V and be a side of P_i. This establishes our claim.

It remains to prove that if P_i has a side s in S_V, then $P_i \in S$. If s is the center of a star in S_V, then by construction P_i is the center of a star in S. If s is a non-center node in some $\text{star}(s') \in S_V$ we distinguish two cases. If s' is a side of some polygon P_j, then P_j intersects P_i and S contains $\text{star}(P_j)$. Consequently, P_i is a non-center node in $\text{star}(P_j)$, thus $P_i \in S$. If s' is a containment segment $x_j x'_j$ where x'_j lies on some polygon P_k, then P_k intersects P_i because $x_j x'_j$ is fully contained in P_k. Because $\text{star}(s') \in S_V$ it follows that $\text{star}(P_k) \in S$, and thus $P_i \in S$. We have reached a contradiction in all cases. We conclude that all sides of P_i are in A_V.

Now consider the case $P_i \in B$. We can follow the proof for the case $P_i \in A$ if the representative side s_i of P_i is in B_V. This must indeed be the case: we cannot have $s_i \in A_V$ because then we would have put P_i into A, and s_i cannot be a node in a star in S_V because then P_i would have been in a star in S, as has been shown above. Hence, the lemma is also true if $P_i \in B$. ◀

We can now prove that there are no edges from A to B in $\mathcal{G}^{\times}[\mathcal{P}]$.

▶ **Lemma 12.** *No polygon $P_a \in A$ intersects any polygon $P_b \in B$.*

Proof. Suppose for a contradiction that P_a and P_b intersect. We distinguish two cases.

- *The boundaries of P_a and P_b intersect.* Let s and s' be the sides of P_a and P_b that intersect. Because $P_a \in A$ we have $s \in A_V$ by Lemma 11. Similarly, we have $s' \in B_V$. But this contradicts that S_V is a separator for $\mathcal{G}^\times[V]$ with parts A_V, B_V.
- *The boundaries of P_a and P_b do not intersect.* Assume wlog that $P_a \subset P_b$. Consider the containment segment $s = x_a x'_a$. By construction of the containment segments, polygon P_b has a side s' that intersects s. Recall that the containment segments were put into the set V_1 that was handled first by the algorithm from the previous section. Hence, there will be a fragment $f \in F$ that is identical to s. Let $i > 1$ be such that $s' \in V_i$. Then f satisfies condition (i) of Lemma 5, with s' playing the role of v. Since s' is a side of $P_b \in B$, we have $s' \in B_V$ by Lemma 11. Thus, Lemma 5 implies that $s = f \in B_V$.
  Similarly there exists a side s'' that is part of P_a that intersects s. From Lemma 11 it follows $s'' \in A_V$. It must be that $s'' \in V_j$ for some $j > 1$. Then f also satisfies condition (i) of Lemma 5 with s'' playing the role of v, so that $s = f \in A_V$. But then the sets A_V and B_V are not disjoint, which contradicts that S_V is a separator. ◀

Putting everything together, we obtain the following theorem. The runtime guarantee for computing the separator is proven in the full version.

▶ **Theorem 13.** *Let P be a set of n constant-complexity c-oriented polygons in the plane. Then the intersection graph $\mathcal{G}^\times[P]$ has a star-based separator of size $O(\sqrt{n})$, which can be computed in $O(n \log n)$ time. Moreover, there is an almost exact distance oracle for $\mathcal{G}^\times[V]$ that uses $O(n\sqrt{n})$ storage and that can report the hop-distance between any two nodes $s, t \in P$, up to an additive error of 2, in $O(\sqrt{n})$ time.*

▶ Remark. The theorem above is stated for n constant-complexity c-oriented polygons. However, it also holds in a more geneal setting, namely for a collection $\mathcal{P}$ of polygons with n edges in total. We can then find a separator of size $O(\sqrt{n})$ and parts A, B such that the number of polygons in A and B is at most $\frac{2|\mathcal{P}|}{3}$. Alternatively, we can guarantee that the total number of edges of the polygons in A (and similarly for B) is at most $\frac{2n}{3}$. Finally, the theorem also works in a weighted setting.

String graphs. A *string graph* is the intersection graph of a set V of curves in the plane [14] – no conditions are put on the curves and, in particular, any two curves in V can intersect arbitrarily many times. (But note that there is still at most one edge between the corresponding nodes in $\mathcal{G}^\times[V]$.) It is known that for any set U of connected regions in the plane, there is a set V of strings such that $\mathcal{G}^\times[U]$ and $\mathcal{G}^\times[V]$ are isomorphic [23]. Thus, string graphs are the most general type of intersection graphs of connected regions in the plane.

Matoušek [25] proved that any string graph with n nodes and m edges has a (node-based) separator of size $O(\sqrt{m} \log m)$.[6] Using this result we can obtain a star-based separator of sublinear size for a string graph $\mathcal{G}^\times[V]$ using the following simple two-stage process.

- *Stage 1:* As long as there is a node $v \in V$ of degree at least $n^{1/3}/\log^{2/3} n$, remove star(v) from $\mathcal{G}^\times[V]$ and add it to the separator. This puts at most $O(n^{2/3} \log^{2/3} n)$ stars into the separator.
- *Stage 2:* Construct a node-based separator on the remaining string graph, using Matoušeks method [25] and put these nodes as singletons into the separator. Since the maximum degree after Stage 1 is $O(n^{1/3}/\log^{2/3} n)$, the remaining graph has $O(n^{4/3}/\log^{2/3} n)$ edges.

[6] A paper by Lee [23] claims that a separator of size $O(\sqrt{m})$ exists for string graphs, but Bonnet et al. [29] note that there is an error in this paper. It is not yet known if the proof can be repaired.

Hence, in Stage 2 we put $O\left(\sqrt{\frac{n^{4/3}}{\log^{2/3} n}} \cdot \log\left(\frac{n^{4/3}}{\log^{2/3} n}\right)\right) = O\left(n^{\frac{2}{3}} \log^{\frac{2}{3}} n\right)$ stars into the separator.

By using the star-based separator exactly as in Section 2, this yields the following.

▶ **Proposition 14.** *Let V be a set of n strings in the plane. Then the string graph $\mathcal{G}^\times[V]$ has a star-based separator of size $O(n^{2/3} \log^{2/3} n)$. Moreover, there is an almost exact distance oracle for $\mathcal{G}^\times[V]$ that uses $O(n^{5/3} \log^{2/3} n)$ storage and that can report the hop-distance between any two nodes $s, t \in V$, up to an additive error of 2, in $O(n^{2/3} \log^{2/3} n)$ time.*

Recall that the distance oracle of Aronov, De Berg and Theocharous [12] for geodesic disks in the plane has $O(n^{7/4+\varepsilon})$ storage and $O(n^{3/4+\varepsilon})$ query time. Thus the distance oracle in Proposition 14 is more general (as it can handle any string graph), has better storage, and better query time. The only downside is that the additive error in our distance oracle is at most 2, while for their oracle it is at most 1.

▶ Remark. One may wonder if any graph, and not just any string graph, admits a sublinear star-based separator, but this is not the case. For example, 3-regular expanders do not admit sublinear node-based separators [13] and working with star-based separators instead of node-based separators does not help in constant-degree graphs.

4 Concluding remarks

Motivated by the fact that intersection graphs of non-fat objects may not admit sublinear node-based or clique-based separators, we introduced a *biclique-based* and *star-based* separators. We proved that the intersection graph of any c-colored set of pseudo-segments has a star-based separator of size $O(\sqrt{n})$, and extended the result to c-oriented polygons. We also presented a straightforward algorithm to compute a star-based separator of size $O(n^{2/3} \log^{2/3} n)$ for any string graph. These results lead to almost exact distance oracles with subquadratic storage and sublinear query times. To the best of our knowledge, such distance oracles did not yet exist – not even for intersection graphs of axis-parallel line segments.

Our work raises several questions. Can we improve the size of star-based separators for string graphs from $O(n^{2/3} \log^{2/3} n)$ to $O(\sqrt{n})$? If not, can we perhaps do so for c-colored sets of strings, or for arbitrary sets of line segments? It is also interesting to explore other applications of biclique-based separators, besides distance oracles, and to see if the bounds we obtained for distance oracles can be improved. While clique-based separators have been used to design subexponential algorithms for problems such as q-Coloring [12] and Dominating Set [11], it is unlikely that our biclique-based separator will yield new results for these problems. This is due to existing $2^{\Omega(n)}$ conditional lower bound (under ETH) for q-Coloring and Dominating Set on 2-dir and segment intersection graphs, respectively [5]. Instead, problems whose main difficulty lies in finding (hop-)distances – computing the diameter in subquadratic time [8] is an example – would be interesting to consider.

Recent work on the (weighted) Independent Set problem on restricted graph classes has exploited properties that can be interpreted through the lens of star-based separators [16, §1.4]. It is known that for every constant t, the family of graphs that does not contain an induced path of length t admits balanced separators that consist of the neighborhoods of $t-1$ vertices (cf. [17, Thm. 1.2]). Hence, such graphs have star-based separators of constant size. This property has been used to develop quasi-polynomial-time approximation schemes for weighted Independent Set on P_t-free graphs [10, 17], as well as exact subexponential-time algorithms for unweighted Independent Set on P_t-free graphs [4]. Can our star-based separators of size $O(\sqrt{n})$ also be used to obtain new algorithms for restricted input families?

References

1 Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004. doi:10.1145/990308.990309.

2 Srinivasa Rao Arikati, Danny Z. Chen, L. Paul Chew, Gautam Das, Michiel H. M. Smid, and Christos D. Zaroliagis. Planar spanners and approximate shortest path queries among obstacles in the plane. In *Proc. 4th Annual European Symposium on Algorithms (ESA 1996)*, pages 514–528, 1996. doi:10.1007/3-540-61680-2_79.

3 Boris Aronov, Mark de Berg, and Leonidas Theocharous. A clique-based separator for intersection graphs of geodesic disks in $\mathbb{R}^2$. In *Proc. 40th International Symposium on Computational Geometry (SoCG)*, pages 9:1–9:15, 2024. doi:10.4230/LIPIcs.SoCG.2024.9.

4 Gábor Bacsó, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Zsolt Tuza, and Erik Jan van Leeuwen. Subexponential-time algorithms for maximum independent set in P_t-free and broom-free graphs. *Algorithmica*, 81(2):421–438, 2019. doi:10.1007/S00453-018-0479-5.

5 Édouard Bonnet and Pawel Rzazewski. Optimality program in segment and string graphs. *Algorithmica*, 81(7):3047–3073, 2019. doi:10.1007/s00453-019-00568-7.

6 Timothy M. Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *J. Comput. Geom.*, 10(2):3–20, 2019. doi:10.20382/JOCG.V10I2A2.

7 Hsien-Chih Chang, Jie Gao, and Hung Le. Computing diameter+1 in truly subquadratic time for unit-disk graphs. *CoRR*, abs/2401.12881, 2024. doi:10.48550/arXiv.2401.12881.

8 Hsien-Chih Chang, Jie Gao, and Hung Le. Computing diameter+2 in truly-subquadratic time for unit-disk graphs. In *Proc. 40th International Symposium on Computational Geometry (SoCG)*, pages 38:1–38:14, 2024. doi:10.4230/LIPICS.SoCG.2024.38.

9 Panagiotis Charalampopoulos, Pawel Gawrychowski, Yaowei Long, Shay Mozes, Seth Pettie, Oren Weimann, and Christian Wulff-Nilsen. Almost optimal exact distance oracles for planar graphs. *J. ACM*, 70(2):12:1–12:50, 2023. doi:10.1145/3580474.

10 Maria Chudnovsky, Marcin Pilipczuk, Michal Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the maximum weight independent set problem in H-free graphs. In *Proc. 31st ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, pages 2260–2278, 2020. doi:10.1137/1.9781611975994.139.

11 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for Exponential-Time-Hypothesis-tight algorithms and lower bounds in geometric intersection graphs. *SIAM J. Comput.*, 49:1291–1331, 2020. doi:10.1137/20M1320870.

12 Mark de Berg, Sándor Kisfaludi-Bak, Morteza Monemizadeh, and Leonidas Theocharous. Clique-based separators for geometric intersection graphs. *Algorithmica*, 85(6):1652–1678, 2023. doi:10.1007/S00453-022-01041-8.

13 Zdeněk Dvořák and Sergey Norin. Strongly sublinear separators and polynomial expansion. *SIAM J. Discrete Math.*, 30(2):1095–1101, 2016. doi:10.1137/15M1017569.

14 Gideon Ehrlich, Shimon Even, and Robert Endre Tarjan. Intersection graphs of curves in the plane. *J. Comb. Theory B*, 21(1):8–20, 1976. doi:10.1016/0095-8956(76)90022-8.

15 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM J. Comput.*, 35(1):151–169, 2005. doi:10.1137/S0097539703436357.

16 Peter Gartland. *Quasi-Polynomial Time Techniques for Independent Set and Beyond in Hereditary Graph Classes*. PhD thesis, University of California, Santa Barbara, USA, 2023. URL: https://www.escholarship.org/uc/item/0kk6d2jv.

17 Peter Gartland, Daniel Lokshtanov, Tomás Masarík, Marcin Pilipczuk, Michal Pilipczuk, and Pawel Rzazewski. Maximum weight independent set in graphs with no long claws in quasi-polynomial time. In *Proc. 56th ACM Symposium on Theory of Computing (STOC 2024)*, pages 683–691, 2024. doi:10.1145/3618260.3649791.

18 I.Ben-Arroyo Hartman, Ilan Newman, and Ran Ziv. On grid intersection graphs. *Discrete Mathematics*, 87(1):41–52, 1991. doi:10.1016/0012-365X(91)90069-E.

19 Petr Hliněný. The maximal clique and colourability of curve contact graphs. *Discrete Applied Mathematics*, 81(1):59–68, 1998. doi:10.1016/S0166-218X(97)00075-9.

20 Jan Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Applied Mathematics*, 52(3):233–252, 1994. doi:10.1016/0166-218X(94)90143-0.

21 Jan Kratochvíl and Jaroslav Nešetřil. Independent set and clique problems in intersection-defined classes of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 031(1):85–93, 1990. URL: http://eudml.org/doc/17810.

22 Hung Le and Christian Wulff-Nilsen. Optimal approximate distance oracle for planar graphs. In *Proc. 62nd IEEE Symposium on Foundations of Computer Science (FOCS 2021)*, pages 363–374. IEEE, 2021. doi:10.1109/FOCS52979.2021.00044.

23 James R. Lee. Separators in region intersection graphs. In *Proc. 8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, pages 1:1–1:8, 2017. doi:10.4230/LIPIcs.ITCS.2017.1.

24 Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. doi:10.1137/0136016.

25 Jirí Matoušek. Near-optimal separators in string graphs. *Comb. Probab. Comput.*, 23(1):135–139, 2014. doi:10.1017/S0963548313000400.

26 Christian Sommer. Shortest-path queries in static networks. *ACM Comput. Surv.*, 46(4):45:1–45:31, 2014. doi:10.1145/2530531.

27 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005. doi:10.1145/1044731.1044732.

28 Jie Xue and Meirav Zehavi. Parameterized algorithms on geometric intersection graphs. *Computer Science Review*, 58:100796, 2025. doi:10.1016/j.cosrev.2025.100796.

29 Édouard Bonnet, Jędrzej Hodor, Tuukka Korhonen, and Tomáš Masařík. Treewidth is polynomial in maximum degree on weakly sparse graphs excluding a planar induced minor. *CoRR*, 2024. arXiv:2312.07962.