Finding Geometric Representations of Apex Graphs is NP-Hard

Dibyayan Chakraborty* and Kshitij Gajjar†

June 3, 2021

Abstract

Planar graphs can be represented as intersection graphs of different types of geometric objects in the plane, e.g., circles (Koebe, 1936), line segments (Chalopin & Gonçalves, 2009), L-shapes (Gonçalves et al., 2018). For general graphs, however, even deciding whether such representations exist is often NP-hard. We consider apex graphs, i.e., graphs that can be made planar by removing one vertex from them. We show, somewhat surprisingly, that deciding whether geometric representations exist for apex graphs is NP-hard.

More precisely, we show that for every positive integer \(k\), recognizing every graph class \(G\) which satisfies \(\text{PURE-2-DIR} \subseteq G \subseteq \text{1-STRING}\) is NP-hard, even when the input graphs are apex graphs of girth at least \(k\). Here, \(\text{PURE-2-DIR}\) is the class of intersection graphs of axis-parallel line segments (where intersections are allowed only between horizontal and vertical segments) and \(\text{1-STRING}\) is the class of intersection graphs of simple curves (where two curves share at most one point) in the plane. This partially answers an open question raised by Kratochvíl & Pergel (2007).

Most known NP-hardness reductions for these problems are from variants of 3-SAT. We reduce from the \(\text{PLANAR HAMILTONIAN PATH COMPLETION}\) problem, which uses the more intuitive notion of planarity. As a result, our proof is much simpler and encapsulates several classes of geometric graphs.

Keywords: Planar graphs, apex graphs, NP-hard, Hamiltonian path completion, recognition problems, geometric intersection graphs, 1-STRING, PURE-2-DIR.

1 Introduction

The recognition a graph class is the decision problem of determining whether a given simple, undirected, unweighted graph belongs to the graph class. Recognition of graph classes is a fundamental research topic in discrete mathematics with applications in VLSI design [CLR83, CLR84, She07]. In particular, when the graph class relates to intersection patterns of geometric objects, the corresponding recognition problem finds usage in disparate areas like map labelling [AVK98], wireless networks [KWZ08], and computational biology [XB06]. The applicability of the recognition problem further increases if the given graph is planar.

Several NP-hard graph problems admit efficient algorithms for planar graphs [Sto73, HW74, Had75]. There are many reasons behind this, owing to the nice structural properties of planar graphs [Tut63, LT79]. Since most of the graphs are non-planar (i.e., the fraction of \(n\)-vertex graphs that are planar approaches zero as \(n\) tends to infinity), it is worthwhile to explore whether these efficient algorithms continue to hold for graphs that are “close to” planar. Specifically, we deal with apex graphs.

Definition 1. [Wel93, TB97]. A graph is an apex graph if it contains a vertex whose removal makes it planar. ♦

Since every \(n\)-vertex apex graph contains an \((n - 1)\)-vertex planar graph as an induced subgraph, one might expect apex graphs to retain some of the properties of planar graphs, in the hope that efficient algorithms for planar graphs carry forward to apex graphs as well. In this paper, we show that for recognition problems for several classes of geometric intersection graphs, this is not the case.

*Indian Institute of Science, Bengaluru. E-mail: dibyayancg@gmail.com
†National University of Singapore, Singapore. A part of this work was done when the author was a postdoctoral researcher at Technion, Israel. E-mail: kshitijgajjar@gmail.com
The intersection graph of a collection $C$ of sets is a graph with vertex set $C$ in which two vertices are adjacent if and only if their corresponding sets have a non-empty intersection. For our main result, we are particularly interested in two natural and well-studied classes of geometric intersection graphs (i.e., when the sets in the collection $C$ are geometric objects in the plane) called PURE-2-DIR and 1-STRING, shown in Figure 1.

**Definition 2.** PURE-2-DIR is the class of all graphs $G$, such that $G$ is the intersection graph of axis-parallel line segments in the plane, where intersections are allowed only between horizontal and vertical segments.

**Definition 3.** 1-STRING is the class of all graphs $G$, such that $G$ is the intersection graph of simple curves in the plane, where two intersecting curves share exactly one point, at which they must cross each other.

**Theorem 4 (Main Result).** Let $g$ be a positive integer and $\mathcal{G}$ be a graph class such that $\text{PURE-2-DIR} \subseteq \mathcal{G} \subseteq \text{1-STRING}$.

Then it is NP-hard to decide whether an input graph belongs to $\mathcal{G}$, even when the input graphs are restricted to bipartite apex graphs of girth at least $g$.

In Figure 1, note that in the PURE-2-DIR (Definition 2) representation, two vertical (similarly, two horizontal) line segments do not intersect each other. Also note that in the 1-STRING (Definition 3) representation, two intersecting curves share exactly one point, and the curves do not touch (tangentially) but instead cross each other at that point. Our main result states that the recognition of every graph class $\mathcal{G}$ that lies between these two is NP-hard, even if the input graph is bipartite, apex, and has large girth. Some candidate graph classes for $\mathcal{G}$ are shown in Subsection 1.1.

**Paper Roadmap:** In Subsection 1.1, we highlight some implications of our main result and in Subsection 1.2, we survey the existing literature on the topic. In Section 2, we describe our proof techniques and give an overview of our proof. In Section 3, we prove Theorem 4 and draw some conclusions in Section 5.

### 1.1 Significance of the Main Result

Our main result has several corollaries, obtained by substituting different values for the graph class $\mathcal{G}$. Recall that the recognition of a graph class $\mathcal{G}$ asks if a given graph $G$ is a member of $\mathcal{G}$.

STRING is the class of intersection graphs of simple curves in the plane. Kratochvíl & Pergel [KP07] posed the question of determining the complexity of recognizing STRING when the inputs are restricted to graphs with large girth. The above question was answered by Mustaţă & Pergel [MP19], where they showed that recognizing STRING is NP-hard, even when the inputs are restricted to graphs of arbitrarily large girth. However, the graphs they constructed were far from planar. Since STRING $\subseteq$ STRING, the following corollary of our main result partially answers Kratochvíl & Pergel’s [KP07] question when the inputs are restricted to apex graphs of large girth.

---

1This illustration is only for representational purposes.

2Formally, a simple curve is a subset of the plane which is homeomorphic to the interval $[0,1]$. 

**Corollary 5.** For every positive integer $g$, recognizing $1$-STRING is NP-hard, even for bipartite apex graphs with girth at least $g$.

Chalopin & Gonçalves [CG09] showed that every planar graph can be represented as an intersection graph of line segments in polynomial time. The following corollary shows that a similar result does not hold for apex graphs.

**Corollary 6.** For every positive integer $g$, recognizing intersection graphs of line segments is NP-hard, even for bipartite apex graphs with girth at least $g$.

Gonçalves, Isenmann & Pennarun [GIP18] showed that every planar graph can be represented as an intersection graph of L-shapes in polynomial time. The following corollary shows that a similar result does not hold for apex graphs.

**Corollary 7.** For every positive integer $g$, recognizing intersection graphs of L-shapes is NP-hard, even for bipartite apex graphs with girth at least $g$.

Our main result also has a connection to a graph invariant called boxicity. The boxicity of a graph is the minimum integer $d$ such that the graph can be represented as an intersection graph of $d$-dimensional axis-parallel boxes. Thomassen showed three decades ago that the boxicity of every planar graph is either one, two or three [Tho86]. It is easy to check if the boxicity of a planar graph is one [BL76]. However, the complexity of determining whether a planar graph has boxicity two or three is not yet known. A result of Hartman, Newman & Ziv [HNZ91] states that the class of bipartite graphs with boxicity 2 is precisely PURE-2-DIR. Combined with our main result, this implies that determining the boxicity of apex graphs is NP-hard.

**Corollary 8.** For every positive integer $g$, recognizing graphs with boxicity 2 is NP-hard, even for bipartite apex graphs with girth at least $g$.

CONV is the class of intersection graphs of convex objects in the plane. Kratochvıl & Pergel [KP07] asked if recognizing CONV remains NP-hard when the inputs are restricted to graphs with large girth. Note that the class of graphs with boxicity 2 (alternatively, intersection graphs of rectangles) is a subclass of CONV. Similarly, intersection graphs of line segments on the plane is also a subclass of CONV. Hence, Corollary 6 and Corollary 8 also address the aforementioned open question of Kratochvıl & Pergel [KP07].

A graph is $c$-apex if it contains a set of $c$ vertices whose removal makes it planar. This is a natural generalization of apex graphs. Our main result implies that no graph class $G$ satisfying PURE-2-DIR $\subseteq G \subseteq 1$-STRING can be recognized in $n^{f(c)}$ time, where $f$ is a computable function depending only on $c$. This means recognizing $G$ is XP-hard, and thus not fixed-parameter tractable [Nie06, DF12] for $c$-apex graphs when parameterized by $c$.

**Corollary 9.** Let $g$ be a positive integer and $G$ be a graph class such that

\[
\text{PURE-2-DIR} \subseteq G \subseteq 1\text{-STRING}.
\]

Then assuming $P \neq NP$, there is no $f(c) \cdot n^{O(1)}$ time algorithm that recognizes $G$ (where $f$ is a computable function depending only on $c$), even for bipartite $c$-apex graphs with girth at least $g$.

Thanks to a long line of work due to Robertson & Seymour [RS04], several graph classes can be characterized by a finite set of forbidden minors. For example, planar graphs are $(K_5, K_{3,3})$-minor free graphs. Interestingly, the set of forbidden minors is not known for apex graphs, although it is known that the set is finite [GI91]. However, it is easy to see that apex graphs are $K_6$-minor free, which means that our main result has the following implication.

**Corollary 10.** Let $g$ be a positive integer and $G$ be a graph class such that

\[
\text{PURE-2-DIR} \subseteq G \subseteq 1\text{-STRING}.
\]

Then it is NP-hard to decide whether an input graph belongs to $G$, even for bipartite $K_6$-minor free graphs with girth at least $g$. 

3
Finally, using techniques different from ours, Kratochvíl & Matoušek [KM89] had shown that recognizing 
\textsc{Pure-2-Dir} is \textsf{NP}-hard, and so is the recognition of line segment intersection graphs. Theorem 4 and Corollary 6 
show that these recognition problems remain \textsf{NP}-hard even when the inputs are restricted to bipartite apex graphs of 
arbitrarily large girth, thereby strengthening their results.

### 1.2 Related Work

In this paper, we focus on recognition problems for subclasses of string graphs. A graph is called a string graph if it is 
the intersection graph of a set of simple curves in the plane. Benzer [Ben59] initiated the study of string graphs 
over half a century ago. Sinden, in his seminal paper [Sin66], asked whether the recognition of string graphs is 
decidable. Kratochvíl [Kra91] took the first steps towards answering Sinden’s question by showing that recognizing 
string graphs is \textsf{NP}-hard. Schaefer, Sedgwick & Štefankovič, [SSŠ03] settled Sinden’s question by showing that 
the recognition of string graphs also lies in \textsf{NP}, and is thus \textsf{NP}-complete.

Finding representations of planar graphs using geometric objects has been a very popular and exciting line of 
research. The celebrated Circle Packing Theorem [Koe36] (also see [And70, Thu82]) states that all planar graphs 
can be expressed as intersection graphs of touching disks. This result also implies that planar graphs are string 
graphs. In his PhD thesis, Scheinerman [Sch84] conjectured that planar graphs can be expressed as intersection 
graphs of line segments. Scheinerman’s conjecture was first shown to be true for bipartite planar graphs [HNZ91], 
and later extended to triangle-free planar graphs [DCCD99]. Chalopin, Gonçalves & Ochem [CGO10] proved a 
relaxed version of Scheinerman’s conjecture by showing that every planar graph is in \textsc{1-String}. Scheinerman’s 
conjecture was finally proved in 2009 by Chalopin & Gonçalves [CG09]. More recently, Gonçalves, Isenmann 
& Pennarun [GIP18] showed that planar graphs are intersection graphs of L-shapes. Soon thereafter, Gonçalves, 
Lévêque & Pinlou [GLP19] showed that planar graphs are also intersection graphs of homothetic triangles.

Another interesting subclass of string graphs is \textsc{k-Dir} [CJ17]. A graph \( G \) is in \textsc{k-Dir} (and has a \textsc{k-Dir} 
representation) if \( G \) is the intersection graph of line segments whose slopes belong to a set of at most \( k \) real numbers. 
Moreover, \( G \) is in \textsc{Pure-k-Dir} if \( G \) has a \textsc{k-Dir} representation where no two segments of the same slope intersect. 
West [Wes91] conjectured that every planar graph is in \textsc{Pure-4-Dir}. A proof of West’s conjecture would have 
provided an alternate proof for the famous Four Color Theorem [AH77] for planar graphs. Indeed, every bipartite 
planar graph is in \textsc{Pure-2-Dir} and every 3-colourable planar graph is in \textsc{Pure-3-Dir} [HNZ91, Gon19]. However, 
Gonçalves [Gon20] proved in 2020 that West’s conjecture is false.

Motivated by the long history of research on geometric intersection representations of planar graphs, we show 
in this paper that finding geometric intersection representations of apex graphs is \textsf{NP}-hard. Our proof technique 
(Section 2) deviates considerably from earlier \textsf{NP}-hardness proofs for recognition of geometric intersection graphs, 
almost of which were based on different variants of 3-SAT. We elaborate on this below.

In a highly influential paper, Kratochvíl [Kra94] introduced a variant of 3-SAT called \textsc{4-Bounded Planar 
3-Connected 3-SAT} (4P3C3SAT), and showed that it is \textsf{NP}-hard. He then used 4P3C3SAT to show that 
recognizing \textsc{Pure-2-Dir} is \textsf{NP}-hard. Chmel [Chm20] reduced the 4P3C3SAT problem to show that recognizing 
intersection graphs of L-shapes is \textsf{NP}-hard. Kratochvíl & Pergel [KP07] used 4P3C3SAT to show that recognizing 
intersection graphs of line segments is \textsf{NP}-hard even if the inputs are restricted to graphs with large girth. Mustaţă 
& Pergel [MP19] also used 4P3C3SAT to generalize and strengthen the above result by showing that there is no 
polynomial time recognizable graph class that lies between \textsc{Pure-2-Dir} and \textsc{String}, even when the inputs are 
restricted to graphs with arbitrarily large girth and maximum degree at most 8. Chaplick, Jelínek, Kratochvíl & 
Vyskočil [CIKV12] used similar techniques to show that the recognition of intersection graphs of rectilinear curves 
having at most \( k \) bends (for every constant \( k \)) is \textsf{NP}-hard.

The construction of the variable and clause gadgets used in these reductions is sometimes quite involved, 
which ends up making the proofs rather complicated. Perhaps the reason behind this is that even though the 
incidence graph (of clauses and variables) of a 4P3C3SAT instance is planar, the variable and clause gadgets are 
non-planar, producing graphs that are far from planar. We overcome this difficulty by reducing from the \textsf{NP}-hard 
\textsc{Planar Hamiltonian Path Completion} problem [AG11], which in turn was inspired by the \textsf{NP}-hard 
\textsc{Planar Hamiltonian Cycle Completion} problem [Wig82]. Reducing from this problem allows us to use 
the natural notion of planarity, making our proofs easier to follow. We explain this in the next section (Section 2).
Figure 2: (a) A front line drawing (Definition 12) of a planar graph $G$, where $G$ is a yes-instance of Planar Hamiltonian Path Completion (Definition 11), and the vertical dotted line denotes the front line $ℓ_{\text{front}}$; (b) $G_{k\div}$ for $k = 3$; (c) $G_{\text{apex}}$; (d) $C$, a Pure-2-DIR representation of $G_{\text{apex}}$ (Subsection 3.3).
2 Proof Techniques

Let us now describe the PLANAR HAMILTONIAN PATH COMPLETION problem [AG11]. A Hamiltonian path in a graph is a path that visits each vertex of the graph exactly once.

**Definition 11.** PLANAR HAMILTONIAN PATH COMPLETION is the following decision problem.

**Input:** A planar graph $G$.

**Output:** Yes, if $G$ is a subgraph of a planar graph with a Hamiltonian path; no, otherwise.

Auer & Gleißner [AG11] showed that this problem (Definition 11) is NP-hard. For their proof, they defined the following constrained planar embedding (Figure 2 (a)).

**Definition 12 (Front line drawing).** A front line drawing of a planar graph $G$ is a planar embedding of $G$ such that its vertices lie on a vertical line segment $\ell_{\text{front}}$ (called the front line by Auer & Gleißner [AG11]) and its edges do not cross each other or $\ell_{\text{front}}$. Let $e = v_i v_j$ be an edge of $G$ such that $e$ connects to the vertex $v_i$ from the left of $\ell_{\text{front}}$, and to the vertex $v_j$ from the right of $\ell_{\text{front}}$. We call such an edge a crossover edge. Figure 2 (a) is a front line drawing with two crossover edges, namely $v_5 v_7$ and $v_4 v_2$. ♦

**Observation 13** (Auer & Gleißner [AG11]). A planar graph $G$ is a yes-instance of the PLANAR HAMILTONIAN PATH COMPLETION problem if and only if $G$ admits a front line drawing.

Using this observation, they showed that deciding whether a graph $G$ admits a front line drawing is NP-hard, implying the required theorem.

**Theorem 14** (Auer & Gleißner [AG11]). PLANAR HAMILTONIAN PATH COMPLETION is NP-hard.

We will use Theorem 14 to show our main result (Theorem 4). We show NP-hardness for graph classes “sandwiched” between two classes of geometric intersection graphs, similar to Mustaţă & Pergel [MP19]. A more technical formulation of our main result is as follows.

**Theorem 15.** For every planar graph $G$ and positive integer $g$, there exists a bipartite apex graph $G_{\text{apex}}$ of girth at least $g$ which can be obtained in polynomial time from $G$, satisfying the following properties.

(a) If $G_{\text{apex}}$ is in 1-STRING, then $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION.

(b) If $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION, then $G_{\text{apex}}$ is in PURE-2-DIR.

**Proof of Theorem 4 assuming Theorem 15.** Let $\mathcal{G}$ be a graph class satisfying the condition PURE-2-DIR $\subseteq \mathcal{G} \subseteq$ 1-STRING, and let $G$ be a planar graph. If $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION, then by Theorem 15 (b), $G_{\text{apex}} \in$ PURE-2-DIR $\subseteq \mathcal{G}$. And if $G_{\text{apex}} \in \mathcal{G} \subseteq$ 1-STRING, then by Theorem 15 (a), $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION.

Thus, $G_{\text{apex}} \in \mathcal{G}$ if and only if $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION. Since PLANAR HAMILTONIAN PATH COMPLETION is NP-hard (Theorem 14) and $G_{\text{apex}}$ can be obtained in polynomial time from $G$, this implies that deciding whether the bipartite apex graph $G_{\text{apex}}$ belongs to $\mathcal{G}$ is NP-hard. □

Therefore, as Theorem 15 implies our main result (Theorem 4), the rest of this paper is devoted to the proof of Theorem 15.

3 Proof of the Main Result

3.1 Construction of the Apex Graph

We begin our proof of Theorem 15 by describing the construction of $G_{\text{apex}}$. Let $G$ be a planar graph. $G_{\text{apex}}$ is constructed in two steps.

$$G \rightarrow G_{k, \text{div}} \rightarrow G_{\text{apex}}.$$  

Let $g \geq 6$ be a positive integer, and $k \geq 3$ be the minimum odd integer greater or equal to $g - 3$. Let $G_{k, \text{div}}$ be the full $k$-subdivision of $G$, i.e., $G_{k, \text{div}}$ is the graph obtained by replacing each edge of $G$ by a path with $k + 1$ edges.
Figure 2 (a) denotes a graph G, and Figure 2 (b) denotes the full 3-subdivision of G. Formally, we replace each $e = (x, y) \in E(G)$ by the path $(x, u_1^e, u_2^e, u_3^e, ..., u_k^e, y)$.

$$
V(G_k_{div}) = V(G) \cup \{u_1^e, u_2^e, u_3^e, ..., u_k^e | e \in E(G)\};
$$

$$
E(G_k_{div}) = \{u_i^e u_{i+1}^e | e \in E(G)\}.
$$

We call the vertices of $V(G) \subseteq V(G_k_{div})$ as the original vertices of $G_k_{div}$ and the remaining vertices as the subdivision vertices of $G_k_{div}$. Finally, we construct $G_{apex}$ by adding a new vertex $a$ to $G_k_{div}$ and making it adjacent to all the original vertices of $G_k_{div}$ (Figure 2 (c)). Formally, $G_{apex}$ is defined as follows.

$$
V(G_{apex}) = V(G_k_{div}) \cup \{a\};
$$

$$
E(G_{apex}) = E(G_k_{div}) \cup \{av | v \in V(G)\}.
$$

Observation 16. If G is a planar graph, then $G_{apex}$ is a bipartite apex graph of girth at least $g$.

Proof. G is a planar graph and subdivision does not affect planarity, so $G_k_{div}$ is also planar, implying that $G_{apex}$ is an apex graph. The vertex set of $G_{apex}$ can be expressed as the disjoint union of two sets $A$ and $B$, where

$$
A = \{x | x \in V(G)\} \cup \{u_i^e | e \in E(G), i \text{ is even}\};
$$

$$
B = \{a\} \cup \{u_i^e | e \in E(G), i \text{ is odd}\}.
$$

Note that $A$ induces an independent set in $G_{apex}$, and so does $B$. Thus, $G_{apex}$ is a bipartite apex graph. As for the girth, note that every cycle in $G_{apex}$ contains at least $k + 2$ vertices $x, u_1^e, u_2^e, u_3^e, ..., u_k^e, y$, for some $e = (x, y) \in E(G)$. At least one more vertex is needed to complete the cycle, implying that the girth of $G_{apex}$ is at least $k + 3 \geq g$.

It is easy to see that this entire construction of $G_{apex}$ from $G$ can be carried out in polynomial time. In Subsection 3.2 and Subsection 3.3, we will prove Theorem 15 (a) and Theorem 15 (b), respectively.

### 3.2 Proof of Theorem 15 (a)

In this section, we will show that if $G_{apex}$ is in 1-STRING, then $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION. In other words, if $G_{apex}$ has a 1-STRING representation, then $G$ is a subgraph of a planar graph with a Hamiltonian path.

In our proofs, we will demonstrate the planarity of our graphs by embedding them in the plane. Typically, a planar graph is defined as a graph whose vertices are points in the plane and edges are strings connecting pairs of points such that no two strings intersect (except possibly at their end points). The same definition holds in more generality, i.e., if the vertices are also allowed to be strings (Figure 3). Let us state this formally.
Definition 17 (Planarizable representation of a graph). A graph $G$ on $n$ vertices and $m$ edges is said to admit a planarizable representation if there are two mutually disjoint sets of strings $V$ and $E$ (with $|V| = n$ and $|E| = m$) in the plane such that

- the strings of $V$ correspond to the vertices of $G$ and the strings of $E$ correspond to the edges of $G$;
- no two strings of $V$ intersect;
- no two strings of $E$ intersect, except possibly at their end points;
- apart from its two end points, a string of $E$ does not intersect any string of $V$;
- for every vertex $v$ and every edge $e = (x, y)$ of $G$, an end point of the string corresponding to $e$ intersects the string corresponding to $v$ if and only if $v = x$ or $v = y$.

Figure 3 illustrates a planar graph and a planarizable representation of it.

Lemma 18. A graph admits a planarizable representation if and only if it is planar.

Lemma 18 may seem obvious. For completeness, we provide a formal proof of it in Section 4. We now use this lemma to prove Theorem 15 (a).

Proof of Theorem 15 (a). Given $G_{apex} \in 1$-STRING, we will show that the planar graph $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION. Let $C$ be a 1-STRING representation of $G_{apex}$ in the plane. It is helpful to follow Figure 4 while reading this proof. We will use $C$ to construct a graph $G_{pl}$ with the following properties.

(a) $G_{pl}$ is a supergraph of $G$ on the same vertex set as $G$.
(b) $G_{pl}$ is planar.
(c) $G_{pl}$ has a Hamiltonian path.

Note that (a), (b), (c) together imply that $G$ is a subgraph of a planar graph with a Hamiltonian path (i.e., $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION). Let $n = |V(G)|$ and assume that $n \geq 4$. Along with our construction of $G_{pl}$, we will also describe $\text{DRAW}(G_{pl})$, a planarizable representation (Definition 17) of $G_{pl}$ in the plane.

In $C$, consider the strings corresponding to the $n$ original vertices (the ones in blue in Figure 4) of $G$. Since the original vertices form an independent set in $G_{apex}$, the blue strings are pairwise disjoint. We add these $n$ strings to $\text{DRAW}(G_{pl})$, which correspond to the $n$ vertices of $G_{pl}$.

Proof of (c): So far, $G_{pl}$ has no edge. We will now add $n - 1$ edges to $G_{pl}$ to connect these vertices via a Hamiltonian path. Recall that each of the $n$ original vertices is adjacent to the apex vertex $a$ in $G_{apex}$, which means that each of the $n$ blue strings intersects $c(a)$ at precisely one point (since $C$ is a 1-STRING representation). Starting from one end point of $c(a)$ and travelling along the curve $c(a)$ until we reach its other end point, we encounter these $n$ points one-by-one. Let $(v_1, v_2, \ldots, v_n)$ be the order in which they are encountered.

For each $i \in [n]$, let $p_i$ be the point at which $c(v_i)$ intersects $c(a)$. For each $i \in [n - 1]$, let $s_i$ be the substring of $c(a)$ between $p_i$ and $p_{i+1}$. Add the strings $s_1, s_2, \ldots, s_{n-1}$ as edges to $\text{DRAW}(G_{pl})$, where $s_1$ represents the edge between $v_1$ and $v_2$. Thus the edges corresponding to the $n - 1$ strings $s_1, s_2, \ldots, s_{n-1}$ constitute a Hamiltonian path $(v_1, v_2, \ldots, v_n)$ in $G_{pl}$. This shows (c).

Proof of (a): To show (a), we need to add all the edges of $G$ to $G_{pl}$ (other than those already added by the previous step), so that $G_{pl}$ becomes a supergraph of $G$. For each edge $e = v_i v_j \in E(G)$, there are $k$ strings $c(u_{e_1}^1), c(u_{e_1}^2), \ldots, c(u_{e_1}^k)$ (corresponding to the subdivision vertices $u_{e_1}^1, u_{e_1}^2, \ldots, u_{e_1}^k$ in $G_{apex}$) in $C$. Note that for each $t \in \{1, 2, \ldots, k\}$, the string $c(u_{e_1}^t)$ intersects exactly two other strings. Let $s_e(u_{e_1}^t)$ be the substring of $c(u_{e_1}^t)$ between those two intersection points. Let $s_e$ be the string obtained by concatenating the $k$ substrings thus obtained.

$$s_e \triangleq \bigcup_{t=1}^{k} s_e(u_{e_1}^t).$$

(19)
Figure 4: (Left) $G_{k\text{-div}} (k=3)$ for a planar graph $G$. (Right) $C$, a planarizable representation of $G_{\text{apex}}$. The bold red string $c(a)$ denotes the apex vertex of $G_{\text{apex}}$. The blue strings denote the original vertices of $G$, and the thin red strings are $c(u_1^2)$, $c(u_2^2)$ and $c(u_3^2)$, corresponding to the subdivision vertices of $G$.

If the edge $e = v_i v_j$ is not already present in $G_{pl}$, then add the string $s_e$ to $\text{DRAW}(G_{pl})$, where $s_e$ represents the edge between $v_i$ and $v_j$ (one end point of $s_e$ lies on $c(v_i)$ and the other on $c(v_j)$). This completes the construction of $\text{DRAW}(G_{pl})$, and shows (a).

**Proof of (b):** To show (b), it is enough to show that $\text{DRAW}(G_{pl})$ is a planarizable representation of $G_{pl}$ (Lemma 18). Note that there are three types of strings in $\text{DRAW}(G_{pl})$: (i) substrings of $c(a)$, (ii) strings of the type $s_e$, for some $e = v_i v_j \in E(G)$, and (iii) $n$ strings corresponding to the original vertices of $G$.

Two strings of type (i) are either disjoint or intersect at their end points, since $c(a)$ is non-self-intersecting. More precisely, for each $i \in [n-1]$, the point $p_{i+1}$ (the unique intersection point of $s_i$, and $s_{i+1}$) lies on $c(v_{i+1})$, which denotes a vertex in $\text{DRAW}(G_{pl})$. A string of type (ii) intersects exactly two strings, $c(v_i)$ and $c(v_j)$, which denote vertices in $\text{DRAW}(G_{pl})$. Finally, strings of type (iii) are mutually disjoint. This shows (b).}

### 3.3 Proof of Theorem 15 (b)

In this section, we will show that if $G$ is a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION, then $G_{\text{apex}}$ is in PURE-2-DIR. In other words, if $G$ is a subgraph of a planar graph with a Hamiltonian path, then $G_{\text{apex}}$ has a PURE-2-DIR representation.

First we elucidate the main idea behind our proof. Given a front line drawing (Figure 2 (a)) of $G$, we will modify it to obtain a PURE-2-DIR representation (Figure 2 (d)) of $G_{\text{apex}}$. The apex segment $c(a)$ takes the place of the front line $l_{\text{front}}$. The edges of $G$, which were strings in the front line drawing, are replaced by rectilinear piecewise linear curves. If we were allowed a large number of rectilinear pieces for each edge, then this construction is trivial, since every curve can be viewed as a series of infinitesimally small vertical and horizontal segments. Our proof formally justifies that this can always be done when the number of allowed rectilinear pieces is a fixed odd integer greater than or equal to three.

**Proof of Theorem 15 (b).** Given $G$, a yes-instance of PLANAR HAMILTONIAN PATH COMPLETION, we will construct $C$, a PURE-2-DIR representation of $G_{\text{apex}}$. Recall that the construction of $G_{\text{apex}}$ from $G$ uses an intermediate graph $G_{k\text{-div}}$, where $k$ is an odd integer greater than or equal to three.

$$G \Rightarrow G_{k\text{-div}} \Rightarrow G_{\text{apex}}.$$

Our proof is by induction on $k$. The major portion of this proof is for $k = 3$. At the end, we will show that if the proof works for $k$, then it also works for $k + 2$, and consequently for all odd integers $k \geq 3$.

**Base case ($k = 3$):** Since $G$ is a subgraph of a planar graph with a Hamiltonian path (Definition 11), $G$ admits a front line drawing (Figure 2 (a)), as per Observation 13. Our construction of $C$ (Figure 2 (d)) is based on and
Let \( n = |V(G)| \) and assume that \( n \geq 4 \). Let \( (v_1, v_2, \ldots, v_n) \) be the ordering of the vertices on the front line \( \ell_{\text{front}} \) from bottom to top. Let \( E_{\text{left}} \) be the set of edges that lie entirely to the left of \( \ell_{\text{front}} \), \( E_{\text{right}} \) be the set of edges that lie entirely to the right of \( \ell_{\text{front}} \), and \( E_{\text{cross}} \) be the set of crossover edges (Definition 12). Further, \( E_{\text{cross}} \) is the disjoint union of \( E_{\text{above}} \) and \( E_{\text{below}} \), where \( E_{\text{above}} \) is the set of crossover edges that go above \( \ell_{\text{front}} \), and \( E_{\text{below}} \) is the set of crossover edges that go below \( \ell_{\text{front}} \). For example, \( v_1 v_2 \in E_{\text{left}}, v_2 v_3 \in E_{\text{right}} \) and \( v_3 v_7 \in E_{\text{below}} \) in Figure 2(a).

Given two points \( p \) and \( q \), we define \( \ell(p, q) \) as the line segment connecting them. Given an edge \( e \), we define \( \text{area}(e) \) as the region enclosed in the closed loop (including the boundary) defined by the union of the edge \( e = v_j v_i \) and the line segment \( \ell(v_j, v_i) \). (Note that \( \ell(v_j, v_i) \) lies on \( \ell_{\text{front}} \).) This notion of area induces a natural partial order \( \preceq \) on the edges of \( G \).

\[ e_1 \preceq e_2 \iff \text{area}(e_1) \subseteq \text{area}(e_2). \]

For example, \( v_2 v_6 \preceq v_1 v_7 \) and \( v_4 v_2 \preceq v_5 v_7 \) in Figure 2(a). It is easy to see that \( \preceq \) is reflexive, anti-symmetric and transitive. Thus, \( (E, \preceq) \) is a poset. Consider the Haase diagram of this poset, where the minimal elements are placed at the bottom and the maximal elements at the top. For an edge \( e \), let \( \text{rank}(e) \) be the number of elements (including \( e \)) on the longest downward chain starting from \( e \). For example, \( \text{rank}(v_2 v_7) = 3 \) because \( v_2 v_5 \preceq v_2 v_6 \preceq v_2 v_7 \) is the longest downward chain starting from \( v_2 v_7 \) in Figure 2(a). Note that the rank of all the minimal elements is one. We are now all set to construct \( C \), our PURE-2-DIR representation of \( G_{\text{apep}} \) for \( k = 3 \) (Figure 2(d)).

**The apex vertex** \( a \): Let \( c(a) \) be the vertical segment \(((0, 0.5), (0, n + 0.5)) \).

**The vertices** \( \{v_1, v_2, \ldots, v_n\} \): For each \( i \in [n] \), let \( c(v_i) \) be the horizontal segment \(((a_i - 0.1, i), (b_i + 0.1, i)) \), where \( a_i \) and \( b_i \) are defined as follows.

\[
\begin{align*}
\alpha_i &= \max\{0\} \cup \{\text{rank}(e) \mid e \text{ is incident to } v_i \text{ and } e \text{ connects to } v_i \text{ from the left of } \ell_{\text{front}}\}; \\
\beta_i &= \max\{0\} \cup \{\text{rank}(e) \mid e \text{ is incident to } v_i \text{ and } e \text{ connects to } v_i \text{ from the right of } \ell_{\text{front}}\}.
\end{align*}
\]

\[ a_i = \max\{\{0\} \cup \{\text{rank}(e) \mid e \text{ is incident to } v_i \text{ and } e \text{ connects to } v_i \text{ from the left of } \ell_{\text{front}}\}\}; \quad (20) \]

\[ b_i = \max\{\{0\} \cup \{\text{rank}(e) \mid e \text{ is incident to } v_i \text{ and } e \text{ connects to } v_i \text{ from the right of } \ell_{\text{front}}\}\}. \quad (21) \]

The \( \{0\} \) set is included to ensure that the argument for the max operation is not an empty set.

**The vertices** \( \{u_1^x, u_2^x, u_3^x\} \): For each edge \( e \) of \( G \), we define a set of four points \( \ell_e = (\alpha_e, \beta_e, \gamma_e, \delta_e) \), such that

\[
\begin{align*}
\alpha_e &= \ell(\alpha_e, \beta_e); \\
\beta_e &= \ell(\gamma_e, \delta_e); \\
\gamma_e &= \ell(\alpha_e, \gamma_e); \\
\delta_e &= \ell(\beta_e, \delta_e).
\end{align*}
\]

We may think of \( \ell_e \) as a piecewise linear curve with three pieces. Let \( \zeta : E \to \{1, 2, \ldots, n^2\} \) be an injective function, i.e., \( \zeta \) maps each edge \( e \) of \( G \) to a distinct number from the set \( \{1, 2, \ldots, n^2\} \). For each edge \( e \), let

\[
\text{xpos}(e) = \text{rank}(e) + \frac{\zeta(e)}{n^2}.
\]

Let \( \varepsilon_k = 1/(k^2 n^3) \) (for this construction, \( k = 3 \)). We use \( \text{xpos} \) and \( \varepsilon_k \) to define the four points of \( \ell_e \), for each \( e \).

\[
\ell_e = \begin{cases} 
\left((-\text{xpos}(e), i), \left((-\text{xpos}(e), \frac{j + 1}{2}\right), \left((-\text{xpos}(e), -\varepsilon_k, \frac{j + 1}{2}\right), (-\text{xpos}(e), -\varepsilon_k, j)) \right) & \text{if } e \in E_{\text{left}}; \\
\left(\left((-\text{xpos}(e), i), \left((-\text{xpos}(e), \frac{j + 1}{2}\right), \left((-\text{xpos}(e), \varepsilon_k, \frac{j + 1}{2}\right), (\text{xpos}(e) + \varepsilon_k, j)) \right) \right) & \text{if } e \in E_{\text{right}}; \\
\left((-\text{xpos}(e), i), (-\text{xpos}(e), n + \text{rank}(e)), ((\text{xpos}(e), n + \text{rank}(e)), (\text{xpos}(e), j)) \right) & \text{if } e \in E_{\text{above}}; \\
\left((-\text{xpos}(e), i), (-\text{xpos}(e), -\text{rank}(e)), ((\text{xpos}(e), -\text{rank}(e)), (\text{xpos}(e), j)) \right) & \text{if } e \in E_{\text{below}}.
\end{cases}
\]

Let \( G_{\text{rep}} \) be the intersection graph of \( C \). We need to show that \( G_{\text{apep}} = G_{\text{rep}} \). First, let us understand the idea behind our construction of \( C \).

\footnote{One way to formulate such a function is by setting \( \zeta(e) = ni + j \) for each \( e = v_ivj \), where \( 1 \leq i < j \leq n \).}
We consider each \( \ell_e \) a single (piecewise linear) segment. Note that in all four cases of its definition, \( \ell_e \) always consists of two vertical segments \((c(u^1_k), c(u^2_k))\) and one horizontal segment \(c(u^0_k)\). Also, the x-coordinate of the vertical segments of \( \ell_e \) is essentially the rank (or the negation of the rank) of \( e \) (Equation 22). The \( \zeta(e)/n^4 \) term (and also the \( \epsilon_k \) term) is simply a tiny perturbation added to its x-coordinate to ensure that the vertical parts of \( \ell_e \) do not intersect the vertical parts of any other \( \ell_{e'} \). (\( \zeta \) was chosen to be an injection for precisely this reason.)

Since \( G_{apex} \) and \( G_{rep} \) have the same vertical set, it is sufficient to show that \( e \in E(G_{apex}) \Leftrightarrow e \in E(G_{rep}) \) in order to demonstrate their equality.

**Proof of** \( e \in E(G_{apex}) \Rightarrow e \in E(G_{rep}) \): The \( c(v_i) \)'s are horizontal segments, all intersecting the vertical apex segment \( c(a) \). Further, the \( c(v_i) \)'s are made to extend as far as (to the left and right of \( c(a) \)) as the maximum rank (plus an additional \( \pm 0.1 \)) of their incident edges (Equation 20, Equation 21). This ensures that they intersect the vertical segments of all their corresponding \( \ell_e \)'s. The fact that the \( c(u^1_k) \)'s and \( c(u^2_k) \)'s intersect their corresponding \( c(u^0_k) \)'s is implicit from the definition of \( \ell_e \).

**Proof of** \( e \notin E(G_{apex}) \Rightarrow e \notin E(G_{rep}) \): Note that \( C \) has three types of segments: (i) the apex \( c(a) \), (ii) the horizontal segments \( c(v_i) \), and (iii) the piecewise linear segments \( \ell_e \). We will consider all pairs of non-adjacent vertices \((p, q)\) of \( G_{apex} \), and show that \( c(p) \) and \( c(q) \) do not intersect in \( C \). We have three cases.

**Case 1:** one of \( c(p) \) or \( c(q) \) is of type (i). Let us say \( c(p) \) is of type (i), i.e., \( p \) is the apex vertex \( a \). Then \( c(p) = ((0,0.5),(0,n+0.5)) \), and \( c(q) \) must be of type (iii) (since all type (ii) vertices are adjacent to \( a \)). Note that the x-coordinates of the vertical pieces of all the \( \ell_e \)'s in \( E_{left} \cup E_{right} \) are either less than \(-0.1\) or greater than \(0.1\), and the y-coordinates of the horizontal pieces of all the \( \ell_e \)'s in \( E_{above} \cup E_{below} \) are either greater than \( n+0.5 \) or less than \(0.5 \). Therefore, \( c(p) \) intersects none of the \( \ell_e \)'s.

**Case 2:** one of \( c(p) \) or \( c(q) \) is of type (ii). Let us say \( c(p) \) is of type (ii). If \( c(q) \) is also of type (ii), then we are done, since all the \( c(v_i) \)'s are mutually disjoint (Equation 20, Equation 21). If \( c(q) \) is of type (iii), then let \( c(q) \) be a piece of \( \ell_e \) for some edge \( e \), and let \( p = v_k \) for some \( k \in [n] \) such that \( e \) is not incident to \( v_k \).

If \( v_k \notin area(e) \), then all segments of \( \ell_e \) lie above \( c(v_k) \) or all segments of \( \ell_e \) lie below \( c(v_k) \), implying that \( \ell_e \) and \( c(v_k) \) do not intersect. If \( v_k \in area(e) \), then \( area(e) \) lies on at least one of the sides (left/right) of \( v_k \in \ell_{front} \). Let \( e' \) be an edge of maximum rank incident to \( v_k \) such that \( e' \) connects to \( v_k \) from the same side (left/right) of \( \ell_{front} \) as \( area(e) \). (If no such edge exists, then the \([0] \) set (Equation 20, Equation 21) comes into play, and we are done, as \( c(v_k) \) falls short of \( \ell_e \).) Recall that \( area(e) \) comprises of \( e \) and the front line \( \ell_{front} \). Using the facts that \( e' \) does not cross \( e \) or \( \ell_{front} \) and that \( e' \) connects to \( v_k \) from the interior of \( area(e) \), we obtain that \( e' \) lies entirely inside \( area(e) \). Therefore,

\[
\text{area}(e') \subseteq \text{area}(e) \Rightarrow e' \leq e \Rightarrow \text{rank}(e') < \text{rank}(e) \Rightarrow \text{rank}(e') + 1 \leq \text{rank}(e).
\]

Note that the vertical pieces of \( \ell_e \) are at least \( \text{rank}(e) \) units away from the apex segment \( c(a) \), and the horizontal segment \( c(v_k) \) only reaches as far as \( \text{rank}(e') + 0.1 < \text{rank}(e) \) units from \( c(a) \) in the direction of \( \ell_e \). Thus, \( c(v_k) \) and \( \ell_e \) do not intersect. Also, all the horizontal pieces of \( \ell_e \) of non-\( \epsilon_k \) length belong to edges of \( E_{cross} \), which lie above \( c(a) \) or below \( c(a) \), and thus none of them intersect \( c(v_k) \).

**Case 3:** both \( c(p) \) and \( c(q) \) are of type (iii). Let \( c(p) \) be a piece of \( \ell_{ep} \) and \( c(q) \) be a piece of \( \ell_{eq} \), for some edges \( e_p \) and \( e_q \) of \( G \). If they are both vertical pieces or one of them is a horizontal piece of length \( \epsilon_k \), then the \( \zeta \) function guarantees that they do not intersect.

Thus, the only remaining case is if one of them (say \( c(p) \)) is a horizontal piece of non-\( \epsilon_k \) length (i.e., \( e_p \) is a crossover edge). Let \( e_p \in E_{above} \) (the proof for \( e_p \in E_{below} \) is similar). The y-coordinate of \( c(p) \) is greater than \( n + 0.5 \). If \( e_q \in E_{left} \cup E_{right} \cup E_{below} \), then all segments of \( \ell_{eq} \) lie below \( c(p) \), and we are done. If \( e_q \in E_{above} \), then note that the all the edges contained in \( E_{above} \) constitute a total order (or chain) in the poset \( (E, \leq) \). Thus either \( \text{rank}(e_p) < \text{rank}(e_q) \) or \( \text{rank}(e_q) < \text{rank}(e_p) \). Let \( \text{rank}(e_p) < \text{rank}(e_q) \) (the proof for \( \text{rank}(e_q) < \text{rank}(e_p) \) is similar). All the pieces of \( \ell_{eq} \) are at least \( \text{rank}(e_p) + 0.5 \) units away from the apex segment \( c(a) \), and all the pieces of \( \ell_{eq} \) reach less than \( \text{rank}(e_p) + 0.5 \) units away from \( c(a) \). Hence, \( \ell_{ep} \) and \( \ell_{eq} \) do not intersect.

This completes the proof of the base case \( (k = 3) \) of our induction. A crucial feature of our construction, which we will exploit in our proof of the inductive case, is that for all edges \( e \) of \( G \), the segment \( c(u^0_k) \) is a vertical segment.
Induction hypothesis: Let \( k \geq 3 \) be an odd integer. Then there exists a PURE-2-DIR representation of \( G_{\text{apex}} \) in which \( c(u_\epsilon^k) \) is a vertical segment for all edges \( e \) of \( G \).

Induction step: Given a PURE-2-DIR representation of \( G_{\text{apex}} \) where \( c(u_\epsilon^k) \) is a vertical segment, we will slightly modify it to include two new segments \( c(u_\epsilon^{k+1}) \) and \( c(u_\epsilon^{k+2}) \) for each \( e \), such that \( c(u_\epsilon^{k+2}) \) is a vertical segment. Let

\[
c(u_\epsilon^k) = ((\lambda_\epsilon^k, \mu_\epsilon^k), (\lambda_\epsilon^k, \pi_\epsilon^k));
\]

\[
\sigma^{k+2}_e = \begin{cases} 
-1 & \text{if } \lambda_\epsilon^k < 0; \\
+1 & \text{if } \lambda_\epsilon^k > 0.
\end{cases}
\]

Recall that \( \epsilon_k = 1/(k^2n^5) \). Now for each edge \( e \) of \( G \), we replace the segment \( c(u_\epsilon^k) \) by the following three segments.

\[
c(u_\epsilon^k) = \left( \left( \lambda_\epsilon^k, \mu_\epsilon^k \right), \left( \lambda_\epsilon^k + \frac{\mu_\epsilon^k + \pi_\epsilon^k}{2} \right) \right);
\]

\[
c(u_\epsilon^{k+1}) = \left( \left( \lambda_\epsilon^k + \frac{\mu_\epsilon^k + \pi_\epsilon^k}{2}, \lambda_\epsilon^k + \sigma^{k+2}_e \epsilon_{k+2} \right), \left( \lambda_\epsilon^k + \frac{\mu_\epsilon^k + \pi_\epsilon^k}{2} \right) \right);
\]

\[
c(u_\epsilon^{k+2}) = \left( \left( \lambda_\epsilon^k + \sigma^{k+2}_e \epsilon_{k+2}, \lambda_\epsilon^k + \frac{\mu_\epsilon^k + \pi_\epsilon^k}{2} \right), \left( \lambda_\epsilon^k + \frac{\mu_\epsilon^k + \pi_\epsilon^k}{2} \right) \right).
\]

Note that these three new segments roughly coincide with the segment that they replaced, with a tiny perturbation of \( \sigma^{k+2}_e \epsilon_{k+2} \) made to the x-coordinates of \( c(u_\epsilon^{k+1}) \) and \( c(u_\epsilon^{k+2}) \). Using the induction hypothesis, it is easy to see that \( c(u_\epsilon^k) \), \( c(u_\epsilon^{k+1}) \) and \( c(u_\epsilon^{k+2}) \) intersect the segments that they are adjacent to in \( G_{\text{apex}} \). The following calculation shows that the \( \sigma^{k+2}_e \epsilon_{k+2} \) perturbation is so minuscule that \( c(u_\epsilon^{k+1}) \) and \( c(u_\epsilon^{k+2}) \) do not intersect any additional segments.

\[
\sum_{i=3}^{k+2} \sigma^i \epsilon_i \leq \frac{1}{n^3} \sum_{i=3}^{k+2} \frac{1}{i^3} < \frac{1}{n^3}.
\]

Note that for every \( e' \neq e \) and every odd \( k' \) such that \( 1 \leq k' \leq k + 2 \), the x-coordinates of \( c(u_\epsilon^{k'}) \) and \( c(u_\epsilon^{k'}) \) differ by roughly \( 1/n^3 \), which is much larger than \( 1/n^5 \). Finally, note that \( c(u_\epsilon^{k+2}) \) is a vertical segment, as promised. This completes the proof. \( \square \)

### 4 Planarizable Representations of Planar Graphs

In this section, we will show Lemma 18, i.e., a graph admits a planarizable representation (Definition 17) if and only if the graph is planar.

**Proof of Lemma 18.** It is easy to see that every planar graph admits a planarizable representation. We will show the other direction: every graph that admits a planarizable representation is planar. Let \( G \) be a graph with a planarizable representation. Let \( v \) be a vertex of \( G \), and \( c(v) \) be its corresponding string. Figure 5 shows the steps of our proof for a given \( c(v) \). Let

\[
R_v = \{ p \mid p \in \mathbb{R}^2, d(p, c(v)) \leq \epsilon \}
\]

be the set of points within a closed \( \epsilon \)-neighbourhood of \( c(v) \), choosing \( \epsilon \) small enough so that \( c(v) \) does not intersect any additional strings. Delete all substrings lying in the interior of \( R_v \). Thus all strings that intersected \( c(v) \) now have one end point on the boundary of \( R_v \). Connect all these boundary end points to a common point (say \( p_v \)) in the interior of \( R_v \) via pairwise disjoint substrings (intersecting only at \( p_v \)) in the interior of \( R_v \), effectively “shrinking” the region \( R_v \) to a single point \( p_v \). (This last step is possible because \( R_v \) is a simply connected region.) Now the point \( p_v \) corresponds to the vertex \( v \).

Do this for all the vertices of \( G \). Since the vertices are now points, and the edges are strings connecting them, the representation thus obtained is a planar drawing of \( G \). \( \square \)
5 Conclusion

Corollary 8 states that recognizing rectangle intersection graphs is NP-hard, even when the inputs are bipartite apex graphs. This raises the following question. Can we recognize planar rectangle intersection graphs in polynomial time? As mentioned earlier, all planar graphs are intersection graphs of 3-dimensional hyper-rectangles (or axis-parallel cuboids) [Tho86, FF11]. Furthermore, there exist series-parallel graphs (a subclass of planar graphs) that are not rectangle intersection graphs [BCR06].

Corollary 9 states that for all graph classes $\mathcal{G}$ such that \textsc{Pure-2-Dir} $\subseteq \mathcal{G} \subseteq \textsc{1-String}$, the recognition of $\mathcal{G}$ is not FPT (fixed-parameter tractable), when parameterized by the apex number of the graph. As our construction produces graphs of large degree, the maximum degree of the graph might be a parameter for which the recognition of $\mathcal{G}$ is FPT.

Corollary 10 states that recognizing several geometric intersection graph classes is NP-hard, even when the inputs are restricted to $K_5$-minor free graphs. On the other hand, the complexity of finding geometric representations of $K_3$-minor free graphs is unknown. Is it possible to use Wagner’s Theorem [Wag37] to decide in polynomial time whether a $K_5$-minor free graph is in \textsc{1-String} (or \textsc{String})? It would also be interesting to study the complexity of recognizing \textsc{String} when the inputs are restricted to apex graphs.

The crossing number of a graph is the minimum number of edge crossings possible in a plane drawing of the graph. Planar graphs are precisely the graphs with crossing number zero. Schaefer showed that apex graphs can have arbitrarily high crossing number, and also exhibited several graphs with crossing number one [Sch18]. Graph classes with a small crossing number, like $k$-planar graphs [ABB+20], have also been studied. Therefore, the complexity of recognizing \textsc{1-String} (or \textsc{String}) when the inputs are restricted to graphs with a small crossing number is another potential direction of research.

Finally, it would be interesting to see if our techniques can be used to prove NP-hardness of recognizing other classes of geometric intersection graphs, like outerstring graphs [BBD18] and intersection graphs of grounded L-shapes [McG96]. Also, the graph classes we study in this paper are for objects embedded in the plane. The complexity of finding geometric intersection representations of apex graphs (appropriately defined) using curves on other surfaces (e.g., torus, projective plane) is another avenue open for exploration.

Acknowledgements: This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 682203-ERC-[Inf-Speed-Tradeoff]. The authors thank the organisers of Graphmasters 2020 [GKR20] for providing the virtual environment that initiated this research.

References

[ABB+20] P. Angelini, M. A. Bekos, F. J. Brandenburg, G. Da Lozzo, G. Di Battista, W. Didimo, M. Hoffmann, G. Liotta, F. Montecchiani, I. Rutter, and C. D. T’Oth. Simple k-planar graphs are simple (k+ 1)-quasiplanar. Journal of Combinatorial Theory, Series B, 142:1–35, 2020.
[AG11] C. Auer and A. Gleißner. Characterizations of deque and queue graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 35–46. Springer, 2011.

[AH77] K. Appel and W. Haken. The solution of the four-color-map problem. *Scientific American*, 237(4):108–121, 1977.

[And70] EM Andreev. On convex polyhedra in lobacevskii spaces. *Mathematics of the USSR-Sbornik*, 10(3):413, 1970.

[AVK98] P. K. Agarwal and M. J. Van Kreveld. *Label placement by maximum independent set in rectangles*. Elsevier, 1998.

[BBD18] T. Biedl, A. Biniaz, and M. Derka. On the size of outer-string representations. In *16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[BCR06] A. Bohra, L. S. Chandran, and J. K. Raju. B-boxicity of series-parallel graphs. *Discrete mathematics*, 306(18):2219–2221, 2006.

[Ben59] S. Benzer. On the topology of the genetic fine structure. *Proceedings of the National Academy of Sciences of the United States of America*, 45(11):1607, 1959.

[BL76] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of computer and system sciences*, 13(3):335–379, 1976.

[CG09] J. Chalopin and D. Gonçalves. Every planar graph is the intersection graph of segments in the plane. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 631–638, 2009.

[CGO10] J. Chalopin, D. Gonçalves, and P. Ochem. Planar graphs have 1-string representations. *Discrete & Computational Geometry*, 43(3):626–647, 2010.

[Chm20] P. Chmel. Algorithmic aspects of intersection representations. *Bachelors Thesis*, 2020.

[CJ17] S. Cabello and M. Jejčič. Refining the hierarchies of classes of geometric intersection graphs. *The Electronic Journal of Combinatorics*, pages P1–33, 2017.

[CJKV12] S. Chaplick, V. Jelínek, J. Kratochvíl, and T. Vyskočil. Bend-bounded path intersection graphs: Sausages, noodles, and waffles on a grill. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 274–285. Springer, 2012.

[CLR83] F. Chung, F. T. Leighton, and A. Rosenberg. *Diogenes: A methodology for designing fault-tolerant VLSI processor arrays*. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Microsystems Program Office, 1983.

[CLR84] F.R.K. Chung, F.T. Leighton, and A.L. Rosenberg. A graph layout problem with applications to VLSI design, 1984.

[DCCD99] N. De Castro, F. J. Cobos, J. C. Dana, A. Mármol, and M. Noy. Triangle-free planar graphs as segments intersection graphs. In *International Symposium on Graph Drawing*, pages 341–350. Springer, 1999.

[DF12] Rodney G Downey and Michael Ralph Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

[FF11] S. Felsner and M. C. Francis. Contact representations of planar graphs with cubes. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 315–320, 2011.
[GI91] Arvind Gupta and Russell Impagliazzo. Computing planar intertwineds. In FOCS, pages 802–811. Citeseer, 1991.

[GIP18] D. Gonçalves, L. Isenmann, and C. Penarun. Planar graphs as L-intersection or L-contact graphs. In Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 172–184. SIAM, 2018.

[GKR20] Leszek Gąsieniec, Ralf Klasing, and Tomasz Radzik. Combinatorial Algorithms: 31st International Workshop, IWOCA 2020, Bordeaux, France, June 8-10, 2020, Proceedings, volume 12126. Springer Nature, 2020.

[GLP19] D. Gonçalves, B. Lévêque, and A. Pinlou. Homothetic triangle representations of planar graphs. Journal of Graph Algorithms and Applications, 23(4):745–753, 2019.

[Gon19] D. Gonçalves. 3-colorable planar graphs have an intersection segment representation using 3 slopes. In International Workshop on Graph-Theoretic Concepts in Computer Science, pages 351–363. Springer, 2019.

[Gon20] D. Gonçalves. Not all planar graphs are in pure-4-dir. Journal of Graph Algorithms and Applications, 24(3):293–301, 2020.

[Had75] Frank Hadlock. Finding a maximum cut of a planar graph in polynomial time. SIAM Journal on Computing, 4(3):221–225, 1975.

[HNZ91] I. B. Hartman, I. Newman, and R. Ziv. On grid intersection graphs. Discrete Mathematics, 87(1):41–52, 1991.

[HW74] John E Hopcroft and Jin-Kue Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In Proceedings of the sixth annual ACM symposium on Theory of computing, pages 172–184, 1974.

[KM89] J. Kratochvíl and J. Matoušek. NP-hardness results for intersection graphs. Commentationes Mathematicae Universitatis Carolinae, 30(4):761–773, 1989.

[Koe36] P. Koebe. Kontaktprobleme der konformen Abbildung. Hirzel, 1936.

[KP07] J. Kratochvíl and M. Pergel. Geometric intersection graphs: do short cycles help? In International Computing and Combinatorics Conference, pages 118–128. Springer, 2007.

[Kra91] J. Kratochvíl. String graphs. II. recognizing string graphs is NP-hard. Journal of Combinatorial Theory, Series B, 52(1):67–78, 1991.

[Kra94] J. Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. Discrete Applied Mathematics, 52(3):233–252, 1994.

[KKWZ08] F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad hoc networks beyond unit disk graphs. Wireless Networks, 14(5):715–729, 2008.

[LT79] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. SIAM Journal on Applied Mathematics, 36(2):177–189, 1979.

[McG96] S. McGuinness. On bounding the chromatic number of L-graphs. Discrete Mathematics, 154(1-3):179–187, 1996.

[MP19] I. Mustaţă and M. Pergel. On unit grid intersection graphs and several other intersection graph classes. Acta Mathematica Universitatis Comenianae, 88(3):967–972, 2019.

[Nie06] Rolf Niedermeier. Invitation to fixed-parameter algorithms. OUP Oxford, 2006.
[RS04] Neil Robertson and Paul D Seymour. Graph minors. xx. wagner’s conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

[Sch84] E. R. Scheinerman. *Intersection classes and multiple intersection parameters of graphs*. Princeton University, 1984.

[Sch18] M. Schaefer. *Crossing numbers of graphs*. CRC Press, 2018.

[She07] N. A. Sherwani. *Algorithms for VLSI Physical Design Automation*. Springer US, 2007.

[Sin66] F. W. Sinden. Topology of thin film rc circuits. *Bell System Technical Journal*, 45(9):1639–1662, 1966.

[SSŠS03] M. Schaefer, E. Sedgwick, and D. Štefankovič. Recognizing string graphs in NP. *Journal of Computer and System Sciences*, 67(2):365–380, 2003.

[Sto73] Larry Stockmeyer. Planar 3-colorability is polynomial complete. *ACM Sigact News*, 5(3):19–25, 1973.

[TB97] D. M. Thilikos and H. L. Bodlaender. Fast partitioning l-apex graphs with applications to approximating maximum induced-subgraph problems. *Information processing letters*, 61(5):227–232, 1997.

[Tho86] C. Thomassen. Interval representations of planar graphs. *Journal of Combinatorial Theory, Series B*, 40(1):9–20, 1986.

[Thu82] William Thurston. Hyperbolic geometry and 3-manifolds. *Low-dimensional topology (Bangor, 1979)*, 48:9–25, 1982.

[Tut63] W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 3(1):743–767, 1963.

[Wag37] K. Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.

[Wel93] D. J. A. Welsh. Knots and braids: some algorithmic questions. *Contemporary Mathematics*, 147, 1993.

[Wes91] D. West. Open problems. *SIAM J. Discrete Math. Newslett.*, 2(1):10–12, 1991.

[Wig82] A. Wigderson. The complexity of the hamiltonian circuit problem for maximal planar graphs. Technical report, Tech. Rep. EECS 198, Princeton University, USA, 1982.

[XB06] J. Xu and B. Berger. Fast and accurate algorithms for protein side-chain packing. *Journal of the ACM (JACM)*, 53(4):533–557, 2006.