FACULTY OF MATHEMATICS AND PHYSICS
Charles University

MASTER THESIS

Petr Chmel

New Intersection Graph Hierarchies

Computer Science Institute of Charles University

Supervisor of the master thesis: doc. RNDr. Vít Jelínek, Ph.D.
Study programme: Computer Science
Study branch: Discrete Models and Algorithms

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to thank doc. RNDr. Vít Jelínek, Ph.D. – the supervisor of this thesis – for all guidance, helpful consultations, and invaluable feedback. I would also like to express gratitude to my family, friends, and especially my girlfriend Míša for supporting me during my studies.

Title: New Intersection Graph Hierarchies

Author: Petr Chmel

Institute: Computer Science Institute of Charles University

Supervisor: doc. RNDr. Vít Jelínek, Ph.D., Computer Science Institute of Charles University

Abstract: String graphs are the intersection graphs of curves in the plane. Asinowski et al. [JGAA 2012] introduced a hierarchy of VPG graphs based on the number of bends and showed that the hierarchy contains precisely all string graphs. A similar hierarchy can be observed with k-string graphs: string graphs with the additional condition that each pair of curves has at most k intersection points. We continue in this direction by introducing precisely-k-string graphs which restrict the representation even more so that each pair of curves has either 0 or precisely k intersection points with all of them being crossings. We prove that for each $k \geq 1$, any precisely-k-string graph is a precisely-$(k+2)$-string graph and that the classes of precisely-k-string graphs and precisely-$(k+1)$-string graphs are incomparable with respect to inclusion.

We also investigate the problem of finding an efficiently representable class of intersection graphs of objects in the plane that contains all graphs with fixed maximum degree. In the process, we introduce a new hierarchy of intersection graphs of unions of d horizontal or vertical line segments, called impure-d-line graphs, and other variations of the class with representation restrictions. We prove that all graphs with maximum degree $\leq 2d$ are impure-d-line graphs and for $d = 1$ this is the best possible. We also study the relationship between the d in the definition of impure-d-line graphs as a parameter and other graph parameters such as treewidth or clique-width.

Keywords: intersection graph, string graph, hierarchy, graph parameter

Contents

Introduction **2**

1 Related works **4**
1.1 Previously studied hierarchies of intersection graphs 4
1.1.1 The motivation 7

2 Unions of line segments **8**
2.1 Properties of the parameters 10
2.2 The parameters and maximum degree 16
2.3 The pure-tile parameter of planar graphs 20
2.4 Relations with other graph parameters 20
2.5 The parameters and the number of vertices 22
2.6 Open problems 23

3 Precise number of intersections **26**
3.1 A version of Euler's formula 26
3.2 Hierarchy inclusions 28
3.3 Hierarchy non-inclusions 29
3.3.1 The Noodle-Forcing Lemma 30
3.3.2 Proofs of the non-inclusions 31
3.4 Open problems 38

Conclusion **40**

Bibliography **41**

List of Figures **45**

Introduction

An intersection graph is a graph such that we can map each vertex v of the graph to a set S_v such that the assigned sets S_u, S_v have a nonempty intersection if and only if the vertices u and v form an edge. The typically investigated classes of intersection graphs are then formed by restricting the properties of the sets S_v. For example, we may restrict the sets to be simple curves in the plane, which yields the class of *string graphs*. The considered classes of graphs will be defined precisely later in Chapter 1.

Quite often, these classes may be restricted by a numerical parameter. These restrictions come in various forms; e.g., in the case of string graphs, the restriction may be in the maximum number of intersections between a pair of curves. A string graph with a representation in which each pair of curves intersects in at most k points is called a k-string graph.

A folklore result also says that the class of string graphs is equivalent to the class of intersection graphs of paths on a rectilinear grid, also abbreviated as VPG. Again, we may restrict the intersection representation further by requiring that the paths have at most k bends, yielding the class B_k-VPG [1]. A similar class arises when considering the class of edge intersection graphs of paths on a rectilinear grid, denoted by EPG. The classes VPG and EPG differ as follows: for a VPG graph, any shared point is enough to form an edge, while two vertices form an edge in an EPG graph if and only if their paths share an edge in the grid. Nevertheless, we may still restrict the paths to have at most k bends, which results in the analogous class B_k-EPG [2].

We can easily observe that the class B_0-EPG is equal to the class of *interval graphs*, the intersection graphs of intervals on a real line [2]. We can use interval graphs to extend the class further by allowing representations that are less restrictive. The first class, so-called *k-interval graphs* are the intersection graphs of unions of at most k disjoint intervals on a single real line – this extends the class of interval graphs by allowing representations with one vertex being represented by more intervals [3, 4]. Another, this time slightly more restrictive, extension of interval graphs is the class of *k-track graphs* [5]. These are the intersection graphs of unions of k intervals on k disjoint parallel real lines (*tracks*) with one interval per track. When mentioning k-track and k-interval graphs, we should note that these classes are relevant in scheduling [6] or bioinformatics [7, 8].

When considering these restrictions by a numerical parameter, we often consider the hierarchy of inclusions of the classes depending on the parameter. In all of the cases mentioned above, increasing the number makes the class larger. Moreover, these inclusions are strict for k-string graphs (proved by Kratochvíl and Matoušek [9]) and B_k-VPG (proved by Chaplick et al. [10]) among other classes.

As with other graph classes, it is customary to consider the complexity of recognition of the class or other decision problems on the class. In this thesis, we only focus on the problem of recognizing the classes. For the mentioned classes, the recognition problem is often NP-complete: k-string graphs for any fixed $k \geq 1$ [11], B_k-VPG for any fixed $k \geq 0$ [11, 10], k-track and k-interval graphs for any fixed $k \geq 2$ [12, 13], B_1-EPG [14] and B_2-EPG [15] with B_k-EPG for any $k \geq 3$ being conjectured to be NP-complete to recognize by Epstein et al. [16] The cases of 1-track, 1-interval, and B_0-EPG graphs are an exception as all of these classes are the same as interval graphs which are recognizable in polynomial time [17, 18]. We also point out that in the case of k-string graphs, the restriction on the number of intersections also quickly yields membership of their recognition in NP, which lies in stark contrast with the case of general string graphs, which were proven to be NP-hard to recognize by Kratochvíl [19] twelve years before Schaefer, Sedgwick, and Štefankovič [20] showed the membership in NP.

In the thesis, we focus on two particular cases of hierarchies of intersection graphs: we first turn to graph classes akin to k-interval and k-track graphs in the plane instead of on the real line, and then we introduce precisely-k-string graphs, which are exactly the k-string graphs such that each pair of curves intersects either in 0 or precisely k points.

In Chapter 1, we review already studied related hierarchies in more detail. Chapter 2 then focuses on new graph classes, called pure-k-tile, pure-k-line, impure-k-tile, and impure-k-line graphs, their relations with each other, and relations with other graph parameters such as treewidth, degeneracy or maximum degree. In Chapter 3, motivated by a version of Euler's formula for intersection graphs of simple curves in the plane, we focus on precisely-k-string graphs, building a hierarchy in which neither of the two inclusions between precisely-k-string graphs and precisely-$(k+1)$-string graphs holds.

Preliminaries

We use standard notation covered by most introductory courses on discrete mathematics. Notably, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$ and $\binom{X}{k}$ to denote the set of all subsets of the set X of size k. We also introduce a few concepts that are often mentioned in the thesis.

A **representation** R of a graph $G = (V, E)$ is a family of sets $R = \{R(v) : v \in V\}$ such that $\forall u, v \in V : R(v) \cap R(u) \neq \emptyset \Leftrightarrow \{u, v\} \in E$.

A **string representation** R of a graph $G = (V, E)$ is a representation $R = \{R(v) : v \in V\}$ such that each $R(v)$ is a piecewise linear curve in the plane.

A string representation is **proper** if every curve is simple, there are finitely many bends and finitely many intersection points, and in every point of the plane, at most two curves intersect, and every such intersection is a crossing. Note that this implies that no two curves are allowed to touch and that an endpoint of a curve cannot lie on another curve.

A **string graph** is an intersection graph of simple curves in the plane. Equivalently, it is an intersection graph of simple piecewise linear curves in the plane.

An **L-shape** is the union of a horizontal line segment and a vertical line segment such that the only common point of the two line segments is the lowest point of the vertical line segment and the leftmost point of the horizontal line segment.

An **L-representation** is a string representation whose every curve is an L-shape.

An **L-graph** is a graph that admits a proper L-representation.

1. Related works

In this chapter, we focus on related works with special focus on hierarchies of intersection graphs. We also mention the motivation for the second chapter; the motivation for the concepts explored in the third chapter follows from an observation in the second chapter.

1.1 Previously studied hierarchies of intersection graphs

We start with three different hierarchies described in one of the initial papers investigating a hierarchy of classes by Kratochvíl and Matoušek [9].

Definition 1 (k-string graph)**.** A *k-string representation* is a proper string representation such that every two curves intersect in at most k points.

A graph is a *k-string graph* if it has a k-string representation.

Definition 2 (k-SEG graph)**.** A *k-SEG representation* is a proper string representation such that each piecewise linear curve consists of at most k segments.

A graph is a *k-SEG graph* if it has a k-SEG representation.

Instead of 1-SEG, we often write just SEG and call the graph a segment graph.

Definition 3 (k-DIR graph)**.** A *k-DIR representation* is a string representation such that each curve consists of a single line segment, and all segments have at most k different slopes.

A graph is a *k-DIR graph* if it has a k-DIR representation.

Definition 4 (PURE-k-DIR graph)**.** A *PURE-k-DIR representation* is a k-DIR representation with the additional constraint that no two parallel line segments intersect. (This is equivalent to saying that it is a proper k-DIR representation.)

A graph is a *PURE-k-DIR graph* if it has a PURE-k-DIR representation.

We can immediately observe that increasing the parameter in these cases always creates a larger class. Moreover, we can also see from the definition that $\forall k \geq 1$: PURE-k-DIR $\subseteq$ k-DIR. Kratochvíl and Matoušek [9] showed these inclusions to be strict while also showing the inclusion of k-SEG graphs in k^2-string graphs. The whole landscape of the hierarchies is as follows: $\forall k \geq 1$: k-DIR $\subsetneq$ $(k+1)$-DIR, PURE-k-DIR $\subsetneq$ PURE-$(k+1)$-DIR, k-SEG $\subsetneq$ $(k+1)$-SEG, k-string $\subsetneq$ $(k+1)$-string, PURE-k-DIR $\subsetneq$ k-DIR, SEG $= \bigcup_{k\geq 0}$PURE-k-DIR$= \bigcup_{k\geq 0} k$-DIR [9].

It is also natural to ask about the complexity of recognizing the classes. Most of these have also been resolved by Kratochvíl and Matoušek [11, 21, 19], showing that the decision problems of recognizing k-string graphs, k-DIR graphs, PURE-k-DIR graphs, all with k fixed, are NP-complete. Moreover, the problem of recognizing SEG graphs in $\exists\mathbb{R}$-complete [9]. (That is, it is polynomially equivalent to the problem of deciding the values of the existential theory of the real numbers. The class $\exists\mathbb{R}$ is defined as the class of all problems that can be reduced in polynomial time to the decision problem of the existential theory of the reals.

The class itself is known to contain NP and to be contained in PSPACE [22]. However, it is an open question whether these inclusions are strict or possibly equalities. For a thorough introduction, we recommend the expository paper on the topic by Matoušek [23].)

An alternative hierarchy of string graphs was introduced by Asinowski et al. [1] In this case, the restriction is similar to k-SEG graphs with the additional constraint that the segments are either horizontal or vertical.

Definition 5 (B_k-VPG and VPG graph)**.** A *B_k-VPG representation* is a string representation such that each piecewise linear curve consists of at most k segments and all of the segments.

A graph is a *B_k-VPG graph* if it has a B_k-VPG representation.

We also define the class VPG $= \bigcup_{k\geq 0} B_k$-VPG.

The abbreviation of the class may seem peculiar at first, meaning "Vertex intersection graphs of Paths on a Grid" which alludes to an alternative view of the representation: we may think of the vertices as being represented by paths on a rectangular grid with two vertices forming an edge if and only if the two respective paths share a vertex. This can be contrasted with the class EPG, "Edge intersection graphs of Paths on a Grid", introduced by Golumbic, Lipshteyn, and Stern [2], which is defined similarly, with the only difference being that two vertices of the graph form an edge if and only if their respective paths share an edge.

Definition 6 (B_k-EPG and EPG graph)**.** A *B_k-EPG representation* of a graph $G = (V, E)$ is a family of piecewise linear curves $R = \{R(v) \colon v \in V\}$ such that $\forall v \in V \colon R(v)$ has at most k bends and every line segment is either horizontal or vertical and $\forall u, v \in V \colon \{u, v\} \in E \Leftrightarrow |R(u) \cap R(v)| = \infty$.

A graph is a *B_k-EPG graph* if it has a B_k-EPG representation.

We also define the class EPG $= \bigcup_{k\geq 0} B_k$-EPG.

Before we focus on the known results about these hierarchies, we first compare the differences between VPG and EPG graphs. The two classes differ wildly: by a folklore result, VPG graphs are precisely string graphs, while Golumbic, Lipshteyn, and Stern [2] showed that every graph is an EPG graph. On the other hand, when comparing the zero-bend classes, B_0-EPG is exactly the class of interval graphs [2], while B_0-VPG is strictly larger as it contains all interval graphs by definition, and additionally, it is easy to observe that the four-cycle C_4 is a B_0-VPG graph but not an interval graph. When comparing hierarchies, we note again that in both cases, increasing the number always yields an inclusion and in all cases, this inclusion is strict. In particular, $\forall k \geq 0$: B_k-VPG$\subsetneq$ B_{k+1}-VPG as shown by Chaplick et al. [10] and $\forall k \geq 0$: B_k-EPG$\subsetneq$$B_{k+1}$-EPG which was first proven by Asinowski and Suk for odd k [24] and later strengthened for all k by Heldt, Knauer, and Ueckerdt [14].

Again, we ask about the complexity of recognizing these classes. The classes B_k-VPG are all NP-complete to recognize as shown by Kratochvíl and Matoušek for $k = 0$ [11] and Chaplick et al. for $k \geq 1$ [10]. Moreover, in the case of a single bend, even if we only have a single rotation of the L-shape at our disposal, the recognition decision problem still remains NP-complete [25, 26]. On the other hand, the only known results for B_k-EPG are restricted to $k \in \{0, 1, 2\}$. The case of B_0-EPG is simple as we can recognize interval graphs in linear time [17, 18]. In the cases of B_1-EPG and B_2-EPG, we know that these classes are NP-complete to recognize as shown by Heldt, Knauer, and Ueckerdt [14] for the case $k = 1$ and Cameron, Chaplick, and Hoàng [27] showed this also extends to all natural subclasses induced by all possible combinations of the four rotations of L-shapes. Pergel and Rzążewski [15] extended these results for the case $k = 2$ and the natural subclasses.

We continue with two closely related hierarchies, both based on extending the class of interval graphs. Interval graphs are often applied to scheduling [28], but often we may wish for a scheduled lecture to have two or more disjoint parts. A similar extension can also be used in bioinformatics for modeling similar regions of DNA sequences [7]. This idea leads to two possible natural definitions. In the first definition, we only expect each vertex to be represented by a union of k intervals on the real line with no other restrictions. The second definition is more restrictive, as in this case, we write the graph as a union of k interval graphs. This is equivalent to a representation where we are given k parallel lines, and each vertex is represented by a union of k intervals, one from each line. We now present a formal definition.

Definition 7 (k-interval graph)**.** A *k-interval representation* of a graph $G = (V, E)$ is a family of k-tuples of intervals on the real line $R = \{(R_1(v), \ldots, R_k(v)) : v \in V\}$ such that $\{u, v\} \in E \Leftrightarrow \exists \ell, m \in \{1, \ldots, k\} : R_\ell(u) \cap R_m(v) \neq \emptyset$.

A graph is a *k-interval graph* if it has a k-interval representation.

Definition 8 (k-track graph)**.** A *k-track representation* of a graph $G = (V, E)$ is a family of k-tuples of intervals $R = \{(R_1(v), \ldots, R_k(v)) \colon v \in V\}$ such that $\forall v \in V, \forall i \in [k] : R_i(v) \subseteq \mathbb{R} \times \{i\}$ and $\{u, v\} \in E \Leftrightarrow \exists \ell \in \{1, \ldots, k\} : R_\ell(u) \cap R_\ell(v) \neq \emptyset$.

A graph is a *k-track graph* if it has a k-track representation.

As with all previous hierarchies, increasing the parameter creates a graph class that is a strict superclass, which was shown for k-interval graphs by Trotter and Harary [3] and for k-track graphs by West and Shmoys [12] who also showed that these two classes are distinct, as for any $k \geq 2$, the complete bipartite graph $K_{k^2+k-1,k+1}$ is a k-interval graph, but not a k-track graph.

Regarding the complexity of recognizing these classes, West and Shmoys [12] also showed that recognizing k-interval graphs for any $k \geq 2$ is NP-complete. Gyarfás and West [5] showed the NP-completeness of recognition of 2-track graphs and conjectured that k-track graphs are NP-complete to recognize for any $k \geq 2$. The conjecture was proved by Jiang [13], who also extended the result to more subclasses such as unit k-track graphs[1]. All these results require $k \geq 2$ because 1-track graphs are the same as 1-interval graphs, and interval graphs can be recognized in linear time, first shown by Booth and Lueker [17] using PQ-trees and later also by Habib et al. [18] using LexBFS.

We may also focus on other possible measures of complexity of a graph representation as the numerical parameter. Cabello and Jejčič [29] consider the classes of k-length segment graphs and k-size-disk graphs. In both cases, the number corresponds to the maximum number of lengths or sizes of the respective objects.

[1] These are k-track graphs with the additional restriction that every interval has unit length.

Definition 9 (k-length-segment graph)**.** A *k-length-segment representation* is a representation such that each curve consists of a single line segment, and all segments have at most k different lengths.

A graph is a *k-length-segment graph* if it has a k-length-segment representation.

Definition 10 (k-size-disk graph)**.** A *k-size-disk representation* of a graph $G = (V, E)$ is family of disks in the plane $R = \{R(v) \subseteq \mathbb{R}^2 : v \in V\}$ such that $\forall u, v \in V : \{u, v\} \in E \Leftrightarrow \exists \ell \in \{1, \ldots, k\} : R_\ell(u) \cap R_\ell(v) \neq \emptyset$ and all the disks in R have at most k distinct diameters.

A graph is a *k-size-disk graph* if it has a k-size-disk representation.

Considering the inclusions, Cabello and Jejčič [29] show that k-length-segment graphs form a strict subclass of $(k+1)$-length-segment graphs for any $k \geq 1$ and the same holds for k-size-disk graphs: for all $k \geq 1$, the class of k-size-disk graphs is strictly contained in the class of $(k+1)$-size-disk graphs.

The complexity of the recognition of these classes seems to be open with the only exception of 1-size-disk graphs, which were shown to be NP-hard to recognize by Breu and Kirkpatrick [30] and later were shown to be $\exists\mathbb{R}$-complete by Kang and Müller [31].

1.1.1 The motivation

The original motivation for Chapter 2 is a paper by Mustaţă and Pergel [32] that considers the decision problem of recognizing classes of intersection graphs with bounded maximum degree. By way of context, we also mention that Kratochvíl and Pergel [33] considered the decision problem of recognizing classes of intersection graphs with large girth.

As Mustaţă and Pergel [32] exhibited classes of intersection graphs that are NP-complete to recognize with bounded degree, it is natural to ask whether there exist classes with bounded degree. In particular, our goal is to find a class of intersection graphs that is trivially recognizable in polynomial time when restricted to graphs with a fixed maximum degree, and it permits a representation that is as simple as possible.

2. Unions of line segments

We are searching for a hierarchy of classes of intersection graphs that satisfy two (conflicting) requirements: we want the representing sets to be as simple as possible, but at the same time, we want to be able to represent as many graphs as possible. To be more precise, our requirement on the simplicity of the sets is formulated via efficient implicit representations.

Definition 11 (Efficient implicit representation [34])**.** An *efficient implicit representation* of a family of graphs $\mathcal{F}$ is an assignment of a $\mathcal{O}(\log n)$-bit code to each vertex of any labeled n-vertex graph $G \in \mathcal{F}$ with an existing polynomial-time decoder depending only on $\mathcal{F}$, i.e., a function that, given two codes of two vertices, computes whether the two vertices are adjacent in the graph or not in polynomial time with respect to the representations.

It is natural to want our representations to be efficient implicit representations. This slightly restricts us as Kannan, Naor, and Rudich [34] observed that any graph class with efficient implicit representation has at most $2^{\mathcal{O}(n \log n)}$ graphs on n vertices. In particular, we cannot hope to represent graphs with either bounded chromatic number or bounded clique number, as there are at least $2^{\Omega(n^2/4)}$ bipartite graphs on n vertices (just by considering all bipartite graphs on n vertices with vertices $1, \dots, \lfloor n/2 \rfloor$ in the first partition and $\lfloor 1 + (n/2) \rfloor, \dots, n$ in the second partition). We, therefore, turn our attention to the classes with a bounded degree or bounded degeneracy.

In particular, we search for a hierarchy of classes of intersection graphs that contain all graphs of maximum degree k or all k-degenerate graphs. As mentioned before, Mustaţă and Pergel [32] showed NP-completeness of recognition of string graphs with maximum degree 8. In contrast, our results will yield classes of intersection graphs that are trivially recognizable in polynomial time when restricted to graphs with maximum degree k.

It is easy to see that for maximum degree 2, even 2-DIR graphs are enough as such graphs are unions of paths and cycles, and both are representable in 2-DIR.

However, a well-known argument by Sinden [35] shows that there exists a bipartite graph of maximum degree 3 that is not a string graph: the subdivision of $K_{3,3}$. Therefore, even for maximum degree 3, we have to resort to a more general model of intersection graphs inspired by k-track and k-interval graphs. In our case, we focus on the case of unions of horizontal and vertical line segments in the plane.

We will use analogous names: for the case where no restrictions are imposed, we call these k-line graphs[1]. In the other case, where we effectively want to represent the graph as a union of k (not necessarily disjoint) intersection graphs, we will use the name k-tile graphs. This naming represents the intuition that we may think of the plane divided into k tiles and in each tile, each vertex is represented by a single line segment.

We may also consider two possibilities with respect to the permitted intersections. Historically, the classes of grid intersection graphs (PURE-2-DIR, sometimes also referred to as bipartite B_0-VPG graphs) and 2-DIR graphs have both

[1] While the name k-segment graphs would be more fitting, it could be easily confused with the class k-SEG.

been considered as the intersection graphs of horizontal and vertical line segments [9]. The only difference is that in PURE-2-DIR, we only permit intersections between a horizontal and a vertical line segment, while in 2-DIR, all intersections are permitted. To easily differentiate between these choices, we explicitly say whether the representations are pure (only intersections between a horizontal and a vertical line segment are allowed) or impure (all intersections are allowed).

These two choices are independent, hence we get four different possible representations.

Definition 12 (Impure-line representation)**.** An *impure-k-line representation* of a graph $G = (V, E)$ is a family of k-tuples of horizontal or vertical line segments $R = \{(R_1(v), \ldots, R_k(v)) : v \in V\}$ such that $\{u, v\} \in E \Leftrightarrow \exists \ell, m \in \{1, \ldots, k\} : R_\ell(u) \cap R_m(v) \neq \emptyset$.

Definition 13 (Impure-tile representation)**.** An *impure-k-tile representation* of a graph $G = (V, E)$ is an impure-k-line representation $R = \{(R_1(v), \ldots, R_k(v)) : v \in V\}$ such that $\forall u, v \in V : \forall \ell, m \in \{1, \ldots, k\} : \ell \neq m \Rightarrow R_\ell(u) \cap R_m(v) = \emptyset$.

Definition 14 (Pure-line representation)**.** A *pure-k-line representation* of a graph $G = (V, E)$ is an impure-k-line representation $R = \{(R_1(v), \ldots, R_k(v)) : v \in V\}$ such that $\forall u, v \in V, \forall \ell, m \in \{1, \ldots, k\}$, if $R_\ell(u) \cap R_m(v) \neq \emptyset$, then precisely one of $R_\ell(u), R_m(v)$ is a horizontal line segment.

Definition 15 (Pure-tile representation)**.** A *pure-k-tile representation* of a graph $G = (V, E)$ is a pure-k-line representation $R = \{(R_1(v), \ldots, R_k(v)) : v \in V\}$ such that $\forall u, v \in V : \forall \ell, m \in \{1, \ldots, k\} : \ell \neq m \Rightarrow R_\ell(u) \cap R_m(v) = \emptyset$.

Definition 16 (Parameters pure-tile, pure-line, impure-tile, impure-line)**.** Given a graph G, we define the following four parameters:

- $\mathrm{il}(G)$ is the least k such that G has an impure-k-line representation,
- $\mathrm{it}(G)$ is the least k such that G has an impure-k-tile representation,
- $\mathrm{pl}(G)$ is the least k such that G has a pure-k-line representation,
- $\mathrm{pt}(G)$ is the least k such that G has a pure-k-tile representation.

Additionally, all of the four representations yield efficient implicit representations whenever k is a fixed constant. In fact, the length of the codes is $4k \log n$ bits for tile versions, as we may assume all coordinates to be at most n and every vertex is represented by k line segments with two endpoints, which in total yields $4k$ coordinates of size $\log n$. In the line versions, we have to be slightly more careful as we can only assume all coordinates to be at most kn, and therefore the codes will be $4k \log(kn)$ bits long.

The polynomial-time decoder for tile versions then looks at the endpoints of the pair of line segments in each tile and checks whether they intersect – if at least one pair intersects, it returns that the two vertices are adjacent, and otherwise, it returns that the two vertices are not adjacent. The line version is slightly slower, as it has to check all k^2 pairs of line segments, but it still runs in polynomial time.

Observation 1. *For any fixed $k \geq 1$, impure-k-line, impure-k-tile, pure-k-line, pure-k-tile graphs all have efficient implicit representations.*

This satisfies our first requirement, and now we show that the second requirement is also satisfied. In particular, we show that all the degeneracy of a graph is an upper bound on its pure-tile parameter, and therefore all d-degenerate graphs are pure-d-tile graphs.

Proposition 2. *Let G be a d-degenerate graph. Then,* $\mathrm{pt}(G) \leq d$.

Proof. We prove this by induction on the number of vertices of G. If $|V(G)| = 1$, we can represent the vertex by a single segment in each of the tiles arbitrarily.

Otherwise, if $|V(G)| > 1$, then we use d-degeneracy and take the vertex $w \in V(G)$ with $\deg_G(w) \leq d$ and by induction on $G - w$, we get a pure-d-tile representation of $G - w$. We denote the neighbors of w in G by $v_1, \ldots, v_\ell$ for $\ell \leq d$. In the i-th tile, we add the new line segment representing w so that it intersects the line segment representing v_i. If $\ell < i \leq d$, we can add the line segment arbitrarily so that it has no intersections. ⊞

Notably, we will show that the classes we described contain graphs that are not string graphs (for $k \geq 2$), which extends the possible range of graphs that can have an efficient implicit representation in an intersection graph class when compared to intersection graph classes which require the objects to be connected.

2.1 Properties of the parameters

Before we tackle the original problem, we start by investigating the properties of the parameters. First, we focus on the relationships with each other and with other well-known parameters.

Observation 3. *For any graph G,* $\mathrm{pt}(G) \geq \mathrm{pl}(G) \geq \mathrm{il}(G)$ *and* $\mathrm{pt}(G) \geq \mathrm{it}(G) \geq \mathrm{il}(G)$

Proof. This follows easily from the fact that any pure representation is also an impure representation and any tile representation is also a line representation. ⊞

Proposition 4. *For any graph G,* $\mathrm{il}(G) \leq \mathrm{pl}(G) \leq 2\mathrm{il}(G)$.

Proof. We only need to prove the second inequality. Let $k := \mathrm{il}(G)$ and let us take an impure-k-line representation R of G.

We may also think of R as a 2-DIR representation of a graph with a larger vertex set $(V(G) \times [k])$, and it is known that 2-DIR $\subseteq$ L-graph [25], hence we get an L-representation of the larger graph. Returning back to the original graph G, we have created a pure-$2k$-line representation of G as each line segment in the original representation was extended into at most two line segments. ⊞

We also note that the constant in the second inequality is the best possible: $\mathrm{it}(K_3) = \mathrm{il}(K_3) = 1$ as we can represent the graph as an interval graph, while $\mathrm{pl}(K_3) = 2$, as all pure-1-line graphs are by definition PURE-2-DIR graphs and, by the characterization by Asinowski et al. [1] under the name of grid intersection graphs, they must be bipartite which K_3 is not. Therefore, $\mathrm{pl}(K_3) \geq 2$, and the pure-2-line representation given by Proposition 4 yields equality.

In particular, Observation 3 and Proposition 4 imply that for a class of graphs $\mathcal{G}$, one of the following four cases must occur.

(I) All four parameters $\mathrm{il}, \mathrm{it}, \mathrm{pl}, \mathrm{pt}$ are bounded.

(II) The parameters $\mathrm{il}, \mathrm{it}, \mathrm{pl}$ are bounded, while pt is unbounded.

(III) The parameters il, pl are bounded, while pt and it are unbounded.

(IV) All four parameters $\mathrm{il}, \mathrm{it}, \mathrm{pl}, \mathrm{pt}$ are unbounded.

We call these classes Type-I to Type-IV respectively. We may easily observe that Type-I and Type-IV classes exist: an example of Type-I class could be the class of graphs with maximum degree 1, as every such graph is a PURE-2-DIR graph, and hence all parameters are equal to one as well. For Type-IV classes, we may consider any class $\mathcal{G}$ that has $2^{\Theta(n^2)}$ graphs on n vertices, as if $\mathcal{G}$ had any of the parameters bounded, it would have an efficient implicit representation. As mentioned at the beginning of this section, Kannan, Naor, and Rudich [34] showed that any graph class with an efficient implicit representation has at most $2^{\mathcal{O}(n \log n)}$ graphs on n vertices, and this would be a contradiction. In fact, we use a similar counting argument in Section 2.5 to show nonconstructive lower bounds on the parameters.

It is natural to ask whether there exist Type-II and Type-III graph classes, and we focus on the question.

We first start by showing a class of graphs that is of Type-II – the class of complete graphs. We will use Turán's theorem during the proof, which we state before moving to our result.

Theorem 5 (Turán [36])**.** *$\forall n \in \mathbb{N}, \forall 2 \leq r \leq n$, the K_{r+1}-free graph with the largest amount of edges is the Turán's graph $T(n,r)$: the complete r-partite graph with partitions of size $\left\lceil \frac{n}{r} \right\rceil$ or $\left\lfloor \frac{n}{r} \right\rfloor$.*

Proposition 6. *Given a complete graph K_n on $n \geq 2$ vertices,* $\mathrm{pt}(K_n) = \lceil \log_2(n) \rceil$.

Proof. We first prove that such representation exists (i.e., we prove the inequality $\mathrm{pt}(K_n) \leq \lceil \log_2(n) \rceil$). Without loss of generality, we may assume that $V(K_n) = \{1, 2, \ldots, n\}$. It is easy to see that any complete bipartite graph can be represented in a single tile. Hence, in each of the $\log_2(n)$ tiles, we represent a certain subgraph as a union of complete bipartite graphs and use induction on the rest of the graph.

If $n = 2$, the statement is obvious as $K_2 \cong K_{1,1}$ and hence it is representable in a single tile.

For the induction step, we assume $n \geq 3$. Then, we use a single tile to represent the largest complete bipartite graph with respect to the number of edges (this follows from Turán's theorem; in fact the weaker Mantel's theorem would suffice [37]) contained in K_n: $G_n = ([n], \{\{a,b\} : 1 \leq a \leq n/2 \wedge n/2 < b \leq n\})$. The graph G_n is complete and bipartite, and, moreover, the edges in G not covered by G_n form a graph that is isomorphic to the disjoint union of a clique

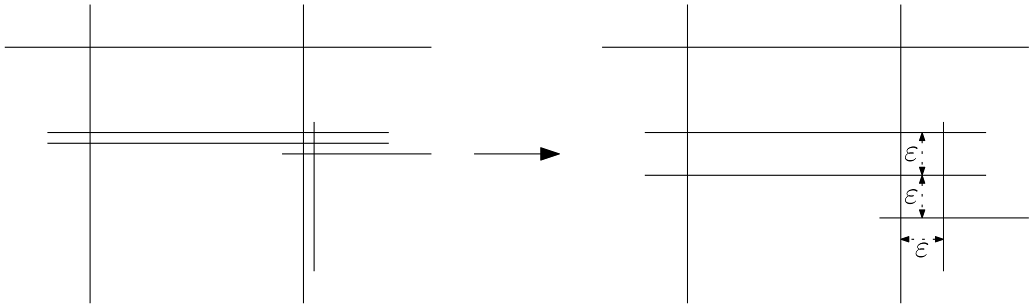

Figure 2.1: The operation of "purifying" the intervals

G' of size $\lfloor n/2 \rfloor$ and a clique G'' of size $\lceil n/2 \rceil$. By the induction hypothesis, we have $\mathrm{pt}(G') \leq \lceil \log_2(\lfloor n/2 \rfloor) \rceil, \mathrm{pt}(G'') \leq \lceil \log_2(\lceil n/2 \rceil) \rceil$.

We then observe that $\lceil \log_2(\lfloor n/2 \rfloor) \rceil \leq \lceil \log_2(n/2) \rceil = \lceil \log_2(n) - \log_2(2) \rceil = \lceil \log_2(n) \rceil - 1$. For the second inequality, we distinguish two cases based on the parity of n. If n is even, then we use the same calculation except with equality: $\lceil \log_2(\lceil n/2 \rceil) \rceil = \lceil \log_2(n/2) \rceil = \lceil \log_2(n) \rceil - 1$. If n is odd, then we calculate $\lceil \log_2(\lceil n/2 \rceil) \rceil = \lceil \log_2((n+1)/2) \rceil = \lceil \log_2(n+1) \rceil - 1$ and since n is odd and we are calculating the logarithm of n in base two, $\lceil \log_2(n) \rceil = \lceil \log_2(n+1) \rceil$.

Therefore, in all cases we need $\lceil \log_2(n) \rceil - 1$ tiles to represent the remaining edges and in total, we use precisely the alotted $\lceil \log_2(n) \rceil$ tiles.

Next, we prove the other inequality showing that no representation with fewer tiles exists, i.e. $\mathrm{pt}(K_n) \geq \lceil \log_2(n) \rceil$. Given the n, let us take the largest $k \in \mathbb{N}$ such that $2^k + 1 \leq n$. We will show that the clique on $2^k + 1$ vertices (which is a subgraph of K_n) has no representation on $k = \lceil \log_2(n) \rceil - 1$ tiles. By contradiction, let us assume such representation exists. Then, each vertex is represented by a horizontal or a vertical line segment in each of the tiles. Let us color the vertices by the k-tuples in $\{H, V\}^k$ where the i-th coordinate denotes whether the vertex is represented by a horizontal or a vertical line segment in the i-th tile. There are only 2^k colors but $2^k + 1$ vertices, and hence two vertices must have the same color. However, this means that the two vertices cannot intersect as in each of the k tiles, they are either both represented by a horizontal line segment or both represented by a vertical line segment. This immediately yields a contradiction, as we found an edge that is not represented, and hence the representation is not correct. ⊞

Observation 7. *Given a complete graph K_n on $n \geq 2$ vertices,* $\mathrm{pl}(K_n) \leq 2$.

Proof. This follows easily from the fact that K_n is an L-graph. ⊞

As all complete graphs are interval graphs, they also have $\mathrm{it}(K_n) = 1, \mathrm{il}(K_n) = 1$. This shows that complete graphs indeed form a Type-II class. Moreover, we show that any Type-II class must have an unbounded clique number.

Proposition 8. *For any graph G,* $\mathrm{it}(G) \leq \mathrm{pt}(G) \leq \omega(G) \cdot \mathrm{it}(G)$.

Proof. Again, only the second inequality is nonobvious.

In this case, let $\ell = \mathrm{it}(G)$ and assume we have an impure-ℓ-tile representation R of G.

We start with taking R and ensuring no two line segments of the same direction have a nonempty intersection by taking all line segments on a single line and spacing them evenly ε-apart for some $\varepsilon > 0$ as shown in Figure 2.1. At the same time, we also appropriately lengthen other line segments so that the only removed intersections are between segments contained in the same line. Note that such segments induce an interval graph.

Finally, we have to remedy the missing intersections. We do this for each tile in parallel, and hence from now on, we focus only on a single tile. We note that these interval graphs in the tile are vertex-disjoint as each vertex is represented by a single line segment and hence has only a single line it lies on in the plane.

By our assumption, we now have an interval graph H with $\omega(H) \leq k$. We observe that any interval graph with $\omega(H) \leq k$ is $(k-1)$-degenerate and by Proposition 2, we know that then, $\mathrm{pt}(H) \leq k-1$.

We can represent all the disjoint interval graphs from the same original tile in the same new "pure" tiles, and the result follows. ⊞

In fact, this also implies a characterization of Type-II classes.

Corollary 9. *If a class of graphs $\mathcal{G}$ has a bounded impure-tile parameter, then it is Type-II if and only if its clique number is unbounded, and it is Type-I otherwise.*

Proof. For a Type-II class, we have the inequality $\mathrm{pt}(G) \leq \omega(G) \cdot \mathrm{it}(G)$, and we know that the impure-tile parameter is bounded, and as the pure-tile parameter is unbounded, the clique number must be unbounded as well. On the other hand, if the clique number is unbounded, the pure-tile parameter cannot be bounded as the graph requires at least $\lceil \log_2(\omega(G)) \rceil$ tiles for the clique by Proposition 6.

If both the impure-tile parameter and the clique number are bounded, then Proposition 8 implies that the pure-tile parameter must be bounded as well. ⊞

Next, we focus on Type-III classes. In particular, we show that the class of multipartite complete graphs is Type-III. For the following proposition, we will employ a Ramsey-type theorem.

Definition 17 (Ramsey number $R(k, c, r)$)**.** For $k, c, r \in \mathbb{N}$, we define the *Ramsey number* $R(k, c, r)$ to be the least integer n such that $n \geq k$ and every set X with n elements has a monochromatic subset[2] of size r with respect to any c-coloring of $\binom{X}{k}$.

Theorem 10 (Ramsey [38]; Theorem 9.1.3 [39])**.** *For all $k, c, r \geq 1$, there exists an $n \geq k$ such that every set X with n elements has a monochromatic subset of size r with respect to any c-coloring of $\binom{X}{k}$. In particular, the Ramsey number $R(k, c, r)$ is well-defined.*

Proposition 11. *For any $K \geq 3$ there exists a graph G_K such that* $\mathrm{it}(G_K) > K$ *and* $\mathrm{pl}(G_K) \leq 2$*. In particular, the impure-tile parameter may be unbounded while pure-line is bounded.*

Proof. For the graph G_K, we construct a particular complete $p(K)$-partite graph $G_K = K_{N(K),\dots,N(K)}$ with $\mathrm{it}(G_K) > K > 2$. We start by observing that any complete k-partite graph is an L-graph and hence it has $\mathrm{pl} \leq 2$. This follows

[2] A subset $Y \subseteq X$ is monochromatic with respect to a c-coloring $\varphi : \binom{X}{k} \to [c]$ of $\binom{X}{k}$ if there exists a color $\ell \in [c] : \forall S \in \binom{Y}{k} : \varphi(S) = \ell$.

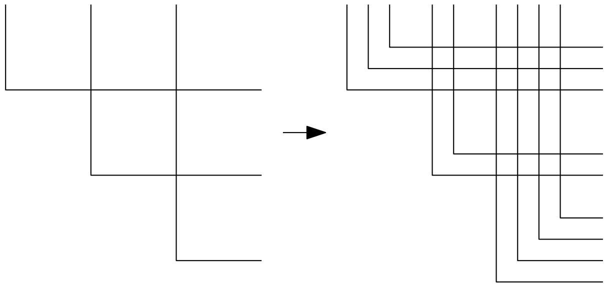

Figure 2.2: Building the L-graph representation of a complete k-partite graph

immediately as we can first represent the complete graph K_k using L-shapes as expected, and then we can blow up each represented vertex of the complete graph into disjoint L-shapes that intersect all other vertices in the graph as shown in Figure 2.2.

We then continue with noticing that taking the complete tripartite graph $K_{2,2,1}$, we have $\mathrm{it}(K_{2,2,1}) > 1$. For contradiction, assume we have an impure-1-tile representation of $K_{2,2,1}$. In the graph, the first two partitions form a C_4. This implies that it cannot happen that all vertices from the first two partitions would be represented on a single line as C_4 is not an interval graph. Moreover, it is easy to see that every 2-DIR representation of C_4 must consist of two parallel horizontal line segments and two parallel vertical line segments with their intersections being the vertices of a rectangle. It immediately follows that the line segment representing the last vertex cannot intersect all of the other line segments in any such representation.

We will construct the complete $p(K)$-partite graph G_K for any fixed $K > 2$ by taking $p(K) = R(2, K, 3)$ (in other words, we know that every K-edge-coloring of a complete graph on $p(K)$ vertices has a monochromatic triangle). We now show that if the partitions are large enough, then there exists a $p(K)$-partite subgraph $S = K_{2,\ldots,2}$ such that the four edges between each pair of partitions are monochromatic. Let all partitions of G_K be of the same size $N = N(K)$ and let v_ℓ^p be the ℓ-th vertex in p-th partition. We consider an auxiliary graph $R = K_N$ that is colored by $K^{p(K)\cdot(p(K)-1)}$ colors with the edge $\{i, j\}$ colored by the $p(K)\cdot(p(K)-1)$-tuple of colors that correspond to the colors of edges $\{v_i^p, v_j^{p'}\}$ over all pairs $(p, p') \in [p(K)] \times [p(K)]$ with $p \neq p'$. For $N = R(2, K^{p(K)(p(K)-1)}, 2p(K))$, we are guaranteed to have a monochromatic clique of size $2p(K)$ in the graph R - we can use these $2p(K)$ vertices $u_1, \ldots, u_{2p(K)}$ to get the required graph by taking the vertices $v^1_{u_1}, v^1_{u_2}, \ldots, v^k_{u_{2k-1}}, v^k_{u_{2k}}, \ldots, v^{p(K)}_{u_{2p(K)-1}}, v^{p(K)}_{u_{2p(K)}}$ with all the edges between partitions monochromatic by the fact that the clique was monochromatic.

By the choice of $p(K)$, there are three partitions such that the edges between them have the same color and therefore, they form an induced monochromatic $K_{2,2,2}$. Therefore, assuming that there is an impure-K-tile representation of the graph G_K, there must be an induced $K_{2,2,2}$ in one tile by setting the colors as tiles in which the edges are represented by the intersection. This is a contradiction with the fact that $\mathrm{it}(K_{2,2,1}) > 1$ as we get an impure-1-tile representation of $K_{2,2,2}$ and $K_{2,2,1}$ is its induced subgraph, yielding an impure-1-tile representation of $K_{2,2,1}$ which is a contradiction, and therefore $\mathrm{it}(G_K) > K > 2$. ⊞

Another example of a Type-III class is the class of triangle-free L-graphs.

Pawlik et al. [40] showed that the class has an unbounded chromatic number. This implies that the class also has an unbounded pure-tile parameter, as for any graph G, we can bound $\chi(G) \leq 2^{\mathrm{pt}(G)}$ by taking a pure-tile representation of G with the least number of tiles and coloring each vertex by a $\{0,1\}$-vector of length $\mathrm{pt}(G)$, where i-th coordinate is 0 if the vertex is represented in i-th tile by a horizontal line segment and the coordinate is 1 if the vertex is represented by a vertical line segment. This is clearly a coloring, as two vertices with the same color cannot form an edge by the definition of the representation, and hence the bound holds. Moreover, the class also has an unbounded impure-tile parameter by Proposition 8, as the class has a bounded clique number and unbounded pure-tile parameter. On the other hand, L-graphs have both pure-line and impure-line parameters bounded by the existence of their L-representation.

Proposition 12. *For any graph G with $\chi(G) \geq 2$, we have* $\mathrm{it}(G) \leq \lceil \log_2(\chi(G)) \rceil \cdot (\mathrm{il}(G))^2$.

Proof. We first start by decomposing the graph G into $\lceil \log_2(\chi(G)) \rceil$ bipartite graphs. Let φ be a $\chi(G)$-coloring of G. As in the proof of Proposition 6, we split the complete graph $K_{\chi(G)}$ into $\lceil \log_2(\chi(G)) \rceil$ bipartite graphs $H_1, \ldots, H_{\lceil \log_2(\chi(G)) \rceil}$ on vertices $[\chi(G)]$. We then blow up each of the graphs H_i into a graph G_i with the vertex set $V(G)$ with two vertices $u, v \in V(G)$ forming an edge in the graph G_i if and only if $\{\varphi(u), \varphi(v)\} \in E(H_i)$ and $\{u, v\} \in E(G)$.

Next, we build an impure-k^2-tile representation from an impure-k-line representation R of a bipartite graph G_i with partitions $A \dot\cup B$. For every pair $(\alpha, \beta) \in [k] \times [k]$, we create a tile that contains the representations $R_\alpha(u)$ for all $u \in A$ and $R_\beta(v)$ for all $v \in B$.

Given an edge $\{u, v\}$ with $u \in A, v \in B$, by definition there exists a pair of line segments $R_a(u), R_b(v)$ such that $R_a(u) \cap R_b(v) \neq \emptyset$. The edge is then realized by the intersection in the tile corresponding to the pair (a, b). Moreover, every pair of vertices that is not connected by an edge has no intersections between any two representing line segments in the impure-k-line representation, and therefore we cannot create the edge in the impure-k^2-tile representation.

We now apply the contruction on each of the $\lceil \log_2(\chi(G)) \rceil$ bipartite graphs G_i, proving the theorem. ⊞

Similarly to Type-II classes, the proposition and the arguments we showed for triangle-free L-graphs imply a characterization of Type-III classes.

Corollary 13. *If a class of graphs $\mathcal{G}$ has bounded impure-line parameter and bounded clique number, then it is Type-III if and only if its chromatic number is unbounded, and it is Type-I otherwise.*

Proof. For a Type-III class, we have the inequality $\mathrm{it}(G) \leq \lceil \log_2(\chi(G)) \rceil \cdot (\mathrm{il}(G))^2$ and we know that the impure-line parameter is bounded and as impure-tile parameter is unbounded, the chromatic number must be unbounded as well.

On the other hand, if the chromatic number of the class is unbounded, the pure-tile parameter must be unbounded as it requires at least $\lceil \log_2(\chi(G)) \rceil$ by the inequality $\chi(G) \leq 2^{\mathrm{pt}(G)}$. Moreover, Proposition 8 implies that as the clique number is bounded, the impure-tile parameter must be unbounded as well, and hence the class is Type-III.

Finally, if the chromatic number of the class is bounded, the impure-tile parameter is bounded by Proposition 12 from the boundedness of both the impure-line parameter and the chromatic number. Corollary 9 then implies that the class must be Type-I, as both the impure-tile parameter and the clique number are bounded. ⊞

2.2 The parameters and maximum degree

As the original motivation is to find a simple class of intersection graphs that contains all graphs with maximum degree d, we now turn to examine the relationship between the parameters and maximum degree.

We can use the 2-factor theorem to obtain similar results to Proposition 2 for graphs with a maximum degree while either getting somewhat worse bounds or weakening our expectations on the requested representation.

Theorem 14 (2-factor theorem, Petersen [41])**.** *If G is a $2k$-regular graph for some $k \in \mathbb{N}$, then the edges of G can be partitioned into k edge-disjoint 2-factors, where a 2-factor is a 2-regular subgraph of G.*

Lemma 15 (Addario-Berry [42])**.** *For a k-partite graph G with maximum degree r, there exists a k-partite r-regular graph G' such that G is an induced subgraph of G'.*

Proof. We prove this by a recursive construction by Addario-Berry [42], which we enhance by showing that the chromatic number stays the same.

We recurse based on the value of $r - \delta(G)$, where $\delta(G)$ is the minimum degree of a vertex in G. If $r - \delta(G) = 0$, we have an r-regular k-partite graph with G as an induced subgraph. Otherwise, we have $r - \delta(G) > 0$. In such case, we take two disjoint copies $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ of G and for any vertex $v \in V(G)$ such that $\deg_G(v) < r$, we add the edge between the corresponding vertices $v_1 \in V_1, v_2 \in V_2$. The resulting graph G' still has the maximum degree r, the minimum degree was raised by one to $\delta(G) + 1$ and hence the difference $r - \delta(G)$ is now smaller.

Moreover, the graph G' is also k-partite: let us have a k-coloring φ of G that we apply to G_1 and G_2, yielding φ_1, φ_2 such that $\forall v \in V(G) : \varphi(v) = \varphi_1(v_1) = \varphi_2(v_2)$. Using these colors together works well in each G_i separately, but the edges connected across have both vertices colored with the same color. We fix this easily by permuting the colors in G_2. Let us take any permutation σ of $[k]$ with no fixed point (for example, we may take a permutation with a single cycle $(1\,2\,\ldots\,k)$, which is possible as $k \geq 2$). We then use the permutation to change the colors on V_2 by setting $\varphi_2'(v) = \sigma(\varphi_2(v))$. This yields the k-coloring ψ of G' as follows:

$$\psi(v) = \begin{cases} \varphi_1(v) & \text{for } v \in V_1, \\ \varphi_2'(v) & \text{for } v \in V_2, \end{cases}$$

Now, all edges in G_1 do not see any change to the colors of their vertices, all edges in G_2 see the colors of both of the vertices change, but as they are only permuted, the two colors remain different, and all edges between G_1 and G_2 now have two different colors, as before, we had $\varphi_1(v_1) = \varphi_2(v_2)$ and now, we have $\varphi_1(v_1) \neq \varphi_2'(v_2)$ and ψ is a k-coloring of G'. ⊞

Theorem 16. *Let G be a bipartite graph with maximum degree $\Delta(G) \leq 2d$ for $d \in \mathbb{N}$. Then* $\operatorname{pt}(G) \leq d$.

Proof. From Lemma 15, we have a $2d$-regular graph G' such that G is an induced subgraph of G' and G' is also bipartite.

We now apply the 2-factor theorem, and we obtain a partition of G' into edge-disjoint 2-factors $F_1, \dots, F_d$. Each of these 2-factors is a union of cycles, and by a characterization of bipartite graphs, a bipartite graph has no odd cycles, and hence each of the cycles in F_i must be an even cycle. It is easy to see that every even cycle has a PURE-2-DIR representation and therefore, we may represent each of these 2-factors in a single pure tile.

In total, we are able to represent the graph in at most d tiles, which yields the claimed bound $\operatorname{pt}(G) \leq d$. ⊞

Theorem 17. *Let G be a graph with maximum degree $\Delta(G) \leq 2d$ for $d \in \mathbb{N}$. Then* $\operatorname{it}(G) \leq d$.

Proof. By Lemma 15, we construct a $2d$-regular graph G' such that G is an induced subgraph of G'. Then, we use the 2-factor theorem to decompose G' into d 2-factors $H_1, \dots, H_d$. It is easy to see that each 2-factor, as a union of disjoint cycles, has a 2-DIR representation. Therefore, we may put each of these representations into a single tile, and we immediately get that $\operatorname{it}(G') \leq d$. Moreover, as G is an induced subgraph of G', we also get that $\operatorname{it}(G) \leq d$. ⊞

We can easily see that for $d = 1$ we cannot represent all graphs with maximum degree ≤ 3 as, by an argument dating back to Sinden [35], the subdivision of $K_{3,3}$ is not a string graph and hence not even a 2-DIR graph. Therefore, the result is tight for $d = 1$.

However, for larger d, our bounds on the maximum degree Δ such that every graph G with $\Delta(G) \leq \Delta$ has $\operatorname{it}(G) \leq d$ differ dramatically. This is mainly caused by the fact that we use Ramsey theory to find a graph that is not even a string graph and such Ramsey graphs tend to be large. To make the bounds slightly better, we extend a lemma by Nešetřil and Rödl [43] for more than two colors while following the proof presented by Diestel in his textbook [39]. While all of these results are constructive in nature, the bounds are rather large, and we show better bounds in Section 2.5 using counting arguments at the expense of being nonconstructive.

We start by stating the necessary tools and definitions.

Definition 18 (Embedding)**.** Given two graphs G, H, an *embedding* of G into H is an injective map $\varphi : V(G) \to V(H)$ such that $\forall u, v \in V(G) : \{u, v\} \in E(G) \Leftrightarrow \{f(u), f(v)\} \in E(H)$.

We remark that this is often called an induced embedding, however as we do not use the usual "subgraph" embedding, we leave out the word "induced" for brevity.

We also recall that we define the Ramsey number $R(k, c, r)$ to be the least integer n such that $n \geq k$ and every set X with n elements has a monochromatic subset of size r with respect to any c-coloring of $\binom{X}{k}$.

Lemma 18 (Nešetřil and Rödl [43]; Lemma 9.3.2 [39]). *Every bipartite graph can be embedded in a bipartite graph of the form* $(X \dot\cup \binom{X}{k}, E)$ *for some set* X *with* $E = \Big\{\{x, Y\} \in X \times \binom{X}{k} : x \in Y\Big\}$.

Lemma 19 (An extension of Lemma 9.3.3. [39], originally by Nešetřil and Rödl [43]). *For every bipartite graph* P *and for every* $\ell \geq 2$*, there exists a bipartite graph* P'_ℓ *such that for every* ℓ*-colouring of the edges of* P'_ℓ*, there is an embedding* $\varphi : P \to P'_\ell$ *for which all the edges of* $\varphi(P)$ *have the same color.*

Proof. We start with the assumption that the graph P has the form from Lemma 18 as otherwise, we can just apply the lemma and search for the supergraph in the right form. Then, we continue by building a graph $P'_\ell := (X' \dot\cup \binom{X'}{k'}, E')$ with $k' := \ell \cdot (k-1) + 1$ and X' is any set of cardinality $|X'| = R(k', \ell\binom{k'}{k}, ((k-1)(\ell-1)+1)|X| + (\ell-1)(k-1))$ with $E' := \Big\{\{x', Y'\} \in X' \times \binom{X'}{k'} : x' \in Y'\Big\}$.

Let P'_ℓ have its edges colored with ℓ colors. When we consider any $Y' \in \binom{X'}{k'}$, it has $k' = \ell \cdot (k-1) + 1$ neighbors, and by the pigeonhole principle, there must exist at least one color c such that at least k edges have the same color. If there are multiple such colors, we choose one of them arbitrarily. We say that Y' is associated to the color via $Z' \subseteq Y'$, where $|Z'| = k$ and $\forall z' \in Z'$, the color of edge $\{z', Y'\}$ is c.

We now assume X' to be linearly ordered. Then for every $Y' \in \binom{X'}{k'}$, a unique order-preserving bijection $\mu_{Y'} \colon Y' \to \{1, \dots, k'\}$ exists. From $\mu_{Y'}$, we get that $\mu_{Y'}[Z'] \in \binom{[k']}{k}$ for Y' associated to its color via Z'. We then color $\binom{X'}{k'}$ by the colors $\binom{[k']}{k} \times [\ell]$ as follows: for a $Y' \in \binom{X'}{k'}$, we set its color to be the pair $(\mu_{Y'}[Z'], j)$, where $j \in [\ell]$ is the color associated with Y' via Z'.

By the choice of $|X'|$ to be the Ramsey number $R(k', \ell\binom{k'}{k}, ((k-1)(\ell-1)+1)|X| + (\ell-1)(k-1))$, we know that X' has a subset W of size $((k-1)(\ell-1)+1)|X| + (\ell-1)(k-1)$, such that $\binom{W}{k'}$ is monochromatic. Moreover, every $Y' \in \binom{W}{k'}$ is associated with the same color α via some Z' (these may differ for different Y'), and the sets Z' lie in their Y' in the same way – in other words, there exists a set $S \in \binom{[k']}{k}$ such that $\mu_{Y'}(Z') = S$ for all $Y' \in \binom{W}{k'}$ and Y' associated with α via Z'.

It now remains to construct the embedding $\varphi : P \to P'_\ell$. We start by defining φ on the set $X = \{x_1, \dots, x_n\}$ by choosing the images $w_i \in W$ so that $\varphi(v_i) = w_i$ and precisely $k' - k = (\ell-1)k + (1-\ell)$ elements of W are smaller than w_1, there are exactly $(\ell-1)k + (1-\ell)$ elements between each pair w_i, w_{i+1} for all $i \in [n-1]$ and there are $(\ell-1)k + (1-\ell)$ elements that are larger than w_n. As we chose the set W so that $|W| = ((k-1)(\ell-1)+1)|X| + (\ell-1)(k-1)$, we have precisely the required number of elements to do so.

The next step is to define φ on the other partition, that is, $\binom{X}{k}$. For $Y \in \binom{X}{k}$, we want to choose $Y' = \varphi(Y) \in \binom{X'}{k'}$ so that the neighbors of Y' among the vertices in $\varphi(X)$ are exactly the images of the neighbors of Y in P. These neighbors are the k vertices $\varphi(x)$ with $x \in Y$, and therefore the colors of the edges are all α.

We find this set Y' by first fixing the subset $Z' := \{\varphi(x) \colon x \in Y\}$ – these correspond to k vertices of the type w_i. We then extend Z' by the remaining

$k' - k$ vertices that are in $W \setminus \varphi[X]$ (that is, by the vertices that are not of the type w_i) so that Z' lies correctly inside Y': $\mu_{Y'}(Z') = S$. We can do this as between any two w_i, there are at least $(\ell - 1)k + (1 - \ell) = k' - k$ other vertices of W.

By definition, we have that $Y' \cap \varphi[X] = Z'$ and hence Y' has the correct neighbors (and non-neighbors). It now remains to notice that φ is injective on $\binom{X}{k}$ as the images Y' of different $Y \in \binom{X}{k}$ must be distinct, as their intersections with X are different and hence also their intersections with $\varphi[X]$ are different. It now follows that $\varphi : P \to P'_\ell$ is indeed an embedding. ⊞

Theorem 20. *For all $d \geq 2$, there exists a bipartite graph G_d with* $\mathrm{it}(G) \geq d$*.*

Proof. For $d = 2$, as observed before, we take the subdivision of $K_{3,3}$, which we denote by $K^2_{3,3}$, to be G_2.

For $d \geq 3$, we take the bipartite graph $G = K^2_{3,3}$ with $\ell = d - 1$, and we apply the Lemma 19. We then get a bipartite graph G'_ℓ.

For contradiction, we assume that $\mathrm{it}(G'_\ell) \leq \ell$. Then, there is an impure-$(d-1)$-tile representation of G. If G'_ℓ can be represented using fewer than ℓ tiles, we can add the remaining tiles so that they have no intersections.

We apply the natural coloring of edges on G'_ℓ, where each edge is colored by the number of a tile in which it is represented. (Note that this coloring is not uniquely determined, as an edge may be represented in multiple tiles; however, in such case, we may choose any of the colors.) By Lemma 19, there exists a color $1 \leq c \leq \ell$ such that G is an induced subgraph of G'_ℓ and all edges of G have the color c. Therefore, in one of the tiles, we have a representation of the graph $K^2_{3,3}$ that is not even a string graph, which is a contradiction.

This implies that G'_ℓ has no impure-tile representation with fewer than d tiles, proving the theorem. ⊞

Corollary 21. *For all $d \geq 2$, there exists a bipartite graph G_d with* $\mathrm{il}(G) \geq d$*.*

Proof. From Proposition 12, we get that for bipartite graphs, we have $\mathrm{it}(G) \leq (\mathrm{il}(G))^2$, implying $\mathrm{il}(G) \geq \sqrt{\mathrm{it}(G)}$. Therefore, given $d \geq 2$ we use the Theorem 20 to get the graph G_{d^2} with $\mathrm{it}(G_{d^2}) \geq d^2$, which immediately implies that $\mathrm{il}(G) \geq d$. ⊞

The bounds in Lemma 19 on the maximum degree are, however, rather large. For the graph $K^2_{3,3}$ and $d \geq 3$, we get th at the size of the ground set in the Ramsey graph is $|X'| = R((d-1) \cdot (k-1) + 1, (d-1) \cdot \binom{(d-1)\cdot(k-1)+1}{k}, ((k-1)(d-2)+1)|X| + (d-2)(k-1))$, where $|X| = 6, k = 2$ as the subdivided vertices in $K^2_{3,3}$ can be thought of as some of the 2-tuples, hence we get $|X'| = R\left(d, (d-1) \cdot \binom{d}{2}, 7(d-1) - 1\right)$. By the definition of G_d, we get that the maximum degree in the graph is $\binom{|X'|-1}{2} = \frac{(|X'|-1)(|X'|-2)}{2}$ – we have two types of vertices, the vertices that correspond to the d-tuples, which have degree $d - 1$ and the vertices corresponding to the elements of X', which have an edge with every 3-tuple containing the element. Therefore, the number of 2-tuples not containing the single element is the maximum degree. Using the stepping-up construction of Erdös, Hajnal, and Rado [44], we have that $R(k, c, r) \geq T_{k-1}(c_1 r^2)$, where $T_k(x)$ is the tower of height k defined inductively as $T_1(x) = x, T_{i+1}(x) = 2^{T_i(x)}$

and c_1 is a positive constant. Therefore, we see that the maximum degree in G_d has a lower bound $(T_{d-1}(c_1 r^2))^2$, which grows much more quickly than the linear growth of the maximum degree of $2d$ of our best impure-d-tile construction.

2.3 The pure-tile parameter of planar graphs

By Proposition 2 and the fact that planar graphs are 5-degenerate, we know that for any planar G, $\mathrm{pt}(G) \leq 5$. However, we can do even better using a result by de Fraysseix, Ossona de Mendez and Pach [45].

Theorem 22 (de Fraysseix, Ossona de Mendez, and Pach [45])**.** *Any bipartite planar graph is a contact intersection graph of horizontal and vertical line segments.*

Theorem 23. *For any planar graph G,* $\mathrm{pt}(G) \leq 2$.

Proof. By the four color theorem, let us have a 4-coloring $\varphi : V(G) \to \{1, 2, 3, 4\}$ of G. We then build two bipartite planar graphs G_1, G_2 so that in the first graph, there are only the edges that have their endpoints colored as either 1 and 3, or 2 and 4. We put all the remaining edges into the second graph. Formally,

- $G_1 = (V(G), \{e \in E(G) : \varphi[e] = \{1, 3\} \vee \varphi[e] = \{2, 4\}\})$,
- $G_2 = (V(G), E(G) \setminus E(G_1))$.

Then, both of these graphs are planar and bipartite, and by Theorem 22, they are contact intersection graphs of horizontal and vertical line segments. Therefore, we have a pure-2-tile representation of G. ⊞

We also immediately notice that this is tight as there are graphs that require two tiles: in particular, the complete graph on four vertices K_4, which by Proposition 6 requires at least two tiles.

2.4 Relations with other graph parameters

In this section, we focus on the relationships of our newly defined parameters with well-known parameters in graph theory. In particular, we focus on treewidth, pathwidth, and clique-width.

We start by defining the parameters and related concepts that will be useful in this section.

Definition 19 (Tree decomposition)**.** A *tree decomposition* of a graph G is a pair (T, X) where T is a tree and $X = \{X_v \in \mathcal{P}(V(G)) \colon v \in V(T)\}$ is a family of so-called *bags* such that

1. $\bigcup X = V(G)$,
2. $\forall e = (u, v) \in E(G) : \exists w \in V(T) : \{u, v\} \subseteq X_w$,
3. $\forall u \in V(G)$: the nodes of T whose bags contain the vertex u form a nonempty connected subgraph (a subtree in particular).

We remark that in order to prevent confusion, it is a custom to say that the tree T has nodes, while the graph G has vertices.

Definition 20 (Treewidth)**.** The *width of a tree decomposition* T is $\max_{X_i \in V(T)} |X_i| - 1$.

The *treewidth* $\mathrm{tw}(G)$ is the minimum possible width of any tree decomposition of G.

Definition 21 (Path decomposition)**.** A *path decomposition* is a tree decomposition (T, X) such that the tree T is a path.

Definition 22 (Pathwidth)**.** The *pathwidth* of a graph G, denoted $\mathrm{pw}(G)$, is the minimum possible width of any path decomposition of G.

Definition 23 (Clique-width)**.** The *clique-width* of a graph G, denoted $\mathrm{cw}(G)$ is the minimum number of labels needed to construct G by using the following four operations:

1. creating a new vertex v with label i (denoted by $i(v)$),
2. taking a disjoint union of two labeled graphs G, H (denoted by $G \oplus H$),
3. given two distinct indices i, j, creating an edge between each vertex with label i and each vertex with label j (denoted by $\eta_{i,j}$),
4. renaming the label i to j (denoted by $\rho_{i \to j}$).

We start by showing that bounded treewidth implies bounded pt parameter.

Proposition 24. *For every graph G,* $\mathrm{pt}(G) \leq \mathrm{tw}(G)$.

Proof. We use induction on the number of vertices to prove that graphs with treewidth at most k are k-degenerate. While this result is not new [46], we prove it for completeness. If $|V(G)| \leq k + 1$, this is obvious as then the maximum degree is at most k, the graph is trivially k-degenerate.

Otherwise, $|V(G)| > k + 1$ and hence any tree decomposition must have at least two bags. This implies that the underlying tree must have a leaf.

Therefore, let us have a leaf node ℓ with its bag X_ℓ and its neighboring node v and its bag X_v. It can happen that $X_\ell \subseteq X_v$, in which case we remove the node ℓ from the tree decomposition and continue by finding a leaf again. Otherwise, $\exists u \in X_\ell : u \notin X_v$ and, as $|X_\ell| \leq k + 1$, $\deg_G(u) \leq k$ and we have found a vertex that is only in the bag X_ℓ and hence has at most k neighbors. Therefore, we found a vertex u of degree $\leq k$ and by the induction hypothesis, $G - \{u\}$ is k-degenerate: it still has treewidth at most k as removing vertices does not increase the treewidth.

We have therefore shown that G is k-degenerate, and by Proposition 2, the conclusion holds. ⊞

Corollary 25. *For every graph G,* $\mathrm{pt}(G) \leq \mathrm{pw}(G)$.

Proof. This follows from the previous proposition since any path decomposition is also a tree decomposition, and hence $\mathrm{tw}(G) \leq \mathrm{pw}(G)$. ⊞

We have shown that if treewidth is bounded, then so is the pure-tile parameter; however, the converse does not hold. It is well-known [47] that the treewidth of an $n \times n$-grid is n; however, it is 2-degenerate, and hence it can be represented in two pure tiles. The same conclusion holds for the pathwidth as for any graph G, $\mathrm{tw}(G) \leq \mathrm{pw}(G)$ which implies that the pathwidth of an $n \times n$-grid is also unbounded.

We now turn our focus to clique-width. In this case, we can only prove partial results. By the definition, it is clear that a complete graph on $n \geq 2$ vertices has clique-width 2, but it requires $\lceil \log_2(n) \rceil$ pure tiles. We, therefore, turn to the slightly stronger parameter pure-line and we take a look at all graphs with clique-width at most 2, so-called *cographs*.

Theorem 26. *For any graph G, if $\mathrm{cw}(G) \leq 2$, then $\mathrm{pl}(G) \leq 2$. Moreover, G is an* L*-graph.*

Proof. The proof follows from two inclusions from the literature. We first use the result by Bose, Buss, and Lubiw [48] which shows that every cograph is a permutation graph. Then, we use the result by Cohen, Golumbic, and Ries [49] which shows that every permutation graph is an L-graph. ⊞

On the other hand, Jelínek [50] showed that the $n \times n$-grid has rank-width precisely $n - 1$ and Oum with Seymour [51] proved that the rank-width of a graph is a lower bound on its clique-width and hence there exist graphs with unbounded clique-width (and rank-width) but bounded pure-line parameter. The question whether bounded clique-width implies bounded pure-line parameter remains open.

2.5 The parameters and the number of vertices

In this section, we focus on bounds on the parameters on n-vertex graphs. First, we show that there exist bipartite n-vertex graphs with impure-line representations requiring $\Omega(n/\log n)$ lines.

Theorem 27. *There exists $n_0 \in \mathbb{N}$ such that $\forall n \geq n_0$, there exists a bipartite graph G with n vertices such that $\mathrm{il}(G) \in \Omega(\frac{n}{\log n})$*

Proof. We do this by upper bounding the number of labeled impure-ℓ-line graphs on n vertices. We create the bound by counting the ways of creating impure-ℓ-line graphs on n vertices: we may take any 2-DIR graph on $\ell \cdot n$ vertices and then split the vertices into n partitions of size ℓ, where each of the partitions corresponds to a single vertex of the graph represented by ℓ line segments.

Therefore, we need to bound the number of 2-DIR graphs on n vertices. We may do this by noting that every 2-DIR graph is an L-graph [25], and there are at most $(n!)^4$ L-representations: every vertex is represented by a single L-shape, and we may order their bends from top to bottom and from left to right, which yields $(n!)^2$. Moreover, we now have to take into account the lengths of the segments, which yields again $(n!)^2$: this follows from the fact that if we assume without loss of generality that the L-shapes are drawn on an $n \times n$-grid, the bottom-most bend point can have its vertical line segment extend into at most n different lengths: from 0 up to $n - 1$, and, for every bend point that is higher, we lose one of the possibilities, which in total yields $n!$ possibilities for the lengths of vertical line segments, and the same for lengths of horizontal line segments. This yields that the number of 2-DIR graphs on ℓn vertices is at most $((\ell n)!)^4 \in 2^{\mathcal{O}(\ell n \log(\ell n))}$. By a result of Scheinerman and Zito [52], this is also asymptotically best possible, as the class of 2-DIR graphs trivially contains all perfect matchings (or, albeit slightly more strongly, all graphs with maximum degree 1).

We can also upper bound the number of ways to split ℓn vertices into n partitions of size ℓ by counting $\prod_{i=0}^{n-1} \binom{\ell(n-i)}{\ell} \leq \binom{\ell n}{\ell}^n \in 2^{\mathcal{O}(\ell n \log(\ell n))}$.

Therefore, we can see that the number of impure-ℓ-line graphs on n vertices is at most $2^{\mathcal{O}(\ell n \log(\ell n))}$.

Moreover, we also easily observe that there are at least $2^{\Omega(\frac{n^2}{4})}$ bipartite graphs as we can simply take (for n even) the bipartite graphs with vertices $1, \ldots, n/2$ in the first partition and the vertices $\frac{n}{2}+1, \ldots, n$ in the second partition. Such graphs are in one-to-one correspondence with 0-1 matrices of size $\frac{n}{2} \times \frac{n}{2}$, and therefore there are at least $2^{\frac{n^2}{4}}$ such labeled graphs.

We are now interested in finding ℓ such that $n^2/4 \in \Theta(\ell n \log(\ell n))$ and we can see that setting $\ell = \Theta(n/\log n)$ fulfills the criterion as then we have $\ell n \log(\ell n) \in \Theta(n^2/4)$. Therefore, for n large enough, there exist bipartite graphs G on n vertices with $\mathrm{il}(G) \in \Omega(\frac{n}{\log n})$, proving the theorem. ⊞

This bound also holds for the three remaining parameters, as we showed the impure-line parameter to be less than or equal to the remaining parameters.

Next, we attempt to complement this result with upper bounds.

Proposition 28. *Every graph on n vertices has a pure-$\left\lceil \frac{n}{\sqrt{2}} \right\rceil$-tile representation.*

Proof. We use the result by Dean, Hutchinson, and Scheinerman [53] that shows that every graph $G = (V, E)$ can be written as a union of $\left\lceil \sqrt{\frac{|E|}{2}} \right\rceil$ trees. As $|E| \leq n^2/2$ for any n-vertex graph and trees are 1-degenerate, by Proposition 2, we can represent each tree as a PURE-2-DIR graph, and therefore we have a pure-tile representation that uses $\left\lceil \sqrt{\frac{|E|}{2}} \right\rceil \leq \left\lceil \sqrt{\frac{n^2}{4}} \right\rceil = \left\lceil \frac{n}{\sqrt{2}} \right\rceil$ tiles. ⊞

Again, this also holds for the three remaining types of representations, as a pure-tile representation is the most restricted type of the four representations we consider.

We remark that Dean, Hutchinson, and Scheinerman [53] also have a result that shows that every graph $G = (V, E)$ can be written as a union of $\left\lfloor \sqrt{\frac{|E|}{3}} + \frac{3}{2} \right\rfloor$ planar graphs. This is a better bound regarding the number of graphs, but to represent a planar graph, we may need two tiles by Theorem 23 and Proposition 6, and therefore, the multiplicative factor we would get is $\frac{2}{\sqrt{6}} > \frac{1}{\sqrt{2}}$ and our bound would be worse.

2.6 Open problems

The first glaring open problem stems from the difference of the bounds on the minimum degree in Lemma 19 and Theorem 17.

Problem. Do there exist graphs with a maximum degree $2d+1, d \geq 2$ such that they do not have an impure-d-tile representation, or does there exist a better lower bound on the minimum degree of a graph that still permits an impure-d-tile representation?

In a similar vein, it would be interesting to investigate the gap between our upper and lower bound of the pure-tile parameter on graphs on n vertices we showed in Theorem 27 and Proposition 28.

Problem. Do there exist graphs on n vertices that require pure-$\Omega(n)$-tile representation or can we find a pure-$f(n)$-tile representation for all n-vertex graphs with $f \in o(n)$?

A third question concerns the relation between bounded clique-width and bounded pure-line parameter.

Problem. Does bounded clique-width imply bounded pure-line parameter?

Finally, it is natural to consider the decision problem of whether a graph G has a pure-d-line representation. We can immediately see that the decision problem is in NP: any n-vertex graph is $(n-1)$-degenerate, and we may therefore assume that $d \leq n$ as the answer is trivial otherwise. Then, there are at most n^2 line segments in the representation and we may assume that the endpoints of the line segments on a $2n^2 \times 2n^2$-grid: take the representation and create a grid so that all endpoints lie on the points of the grid. As there are at most $2n^2$ endpoints and each of them may create a new grid line both horizontally and vertically, we get to the size $2n^2 \times 2n^2$. (In fact, each line segment creates at most three grid lines, one that the line segment lies on and two that are perpendicular to the line segment and intersect it in the endpoints. However, for the membership in NP, the above bound is sufficient.)

It is now obvious that we can use the description of the whole representation as a polynomial certificate, as each of the n^2 line segments requires only $2\log_2(2n^2)$ bits and hence the polynomial certificate has size at most $2n^2 \cdot \log_2(2n^2)$. The verification of the certificate then can be done in time $\mathcal{O}(n^4)$ as we can check every pair of vertices and all $\binom{n}{2}$ pairs of representation line segments representing the vertices to either intersect or not intersect.

The case of $d = 1$ has been resolved by Kratochvíl [21] who showed that the decision problem is NP-complete and the same also holds for other three variants of the representations in the case of $d = 1$. Moreover, the decision problem of recognizing impure-d-tile graphs with maximum degree $2d$ is trivially in P by Theorem 17 and recognizing pure-d-tile graphs with maximum degree d is also in P by Proposition 2.

The problem has already been studied with a similar parameter called thickness, that is, the minimum number of planar graphs such that G can be written as their union. In this case, Mansfield [54] showed that deciding whether thickness is at most k for any $k \geq 2$ is NP-complete. However, adapting such reduction for the case of intersection representations proved to be tricky, as a key component in the reduction was the ability to have a linear bound on the number of edges in a planar graph that comes from Euler's formula. While we can prove a version of Euler's formula for special cases of intersection representations and we do so in the next chapter, it does not yield any nontrivial bound on the number of edges unless we also require a large girth of the graph, which was first used by Nešetřil and Kostochka [55]. This is a consequence of the fact that in planar graphs, edges are represented by curves, while in intersection representations (and particularly proper representations), edges are represented by the intersections, that is, the points.

Problem. What is the complexity of deciding whether a general graph G has a pure-d-line representation for $d \geq 2$? The same question also applies to pure-d-tile, impure-d-line, and impure-d-tile.

3. Precise number of intersections

In this chapter, we focus on a parameter that has not been studied in much detail: the number of intersections of string graphs. While 1-string graphs have been studied in the past, the subclasses of string graphs with a higher number of intersections have not been in the spotlight, with a few exceptions.

We study these classes, mainly motivated by a version of Euler's formula for intersection graphs in the plane. While the method used in the proof was first used by Nešetřil and Kostochka [55] in proving bounds on chromatic numbers of 1-string graphs with large girth, they did not explicitly state the formula – Euler's formula is mentioned in the process, however, they use it on the "graphical graph" $H(F)$ which is planar.

3.1 A version of Euler's formula

Theorem 29 (Euler's formula for 1-string graphs)**.** *Let G be a connected 1-string graph with a 1-string representation R. Let f denote the number of faces of R.*

Then, $|V(G)| - |E(G)| + f = 2$.

Proof. We use the same terminology as Nešetřil and Kostochka: given the proper 1-string representation R of a graph G, we construct a plane graph $H(R)$ (called the graphical graph) as follows. The vertices of $H(R)$ are precisely the intersection points of the curves. Two vertices of $H(R)$ are joined by an edge if and only if they lie on the same curve and there is no other intersection point in between them. The drawing of $H(R)$ immediately follows from the definition: the vertices are drawn in the place of the intersection points and the edges are drawn on the curve joining the intersection points.

Immediately, we see that $|V(H(R))| = |E(G)|$ as each pair of curves may intersect at most once, and hence each edge is counted precisely once. Moreover, $|E(H(R))| = 2|E(G)| - |V(G)|$ as on each curve $R(v)$, there are precisely $\deg_G(v)$ intersection points and hence $\deg_G(v) - 1$ edges. In total, we get $\sum_{v \in V(G)}(\deg_G(v) - 1) = 2|E(G)| - |V(G)|$. We also immediately observe that the drawing of $H(R)$ has the same number of faces as the representation R, we denote the number by f. By Euler's formula, we know that $2 = |V(H(R))| - |E(H(R))| + f = |E(G)| - (2|E(G)| - |V(G)|) + f = |V(G)| - |E(G)| + f$, proving the theorem. ⊞

We can also extend this to string graphs with at most k intersection points per each pair of curves. If we know that all pairs of strings have precisely k intersections, the inequality in the formula will then become equality. Therefore, we also define a subclass of k-string graphs in which we require that the number of intersections is always either k or zero.

Definition 24 (Intersection point)**.** An *intersection point* of a string representation R is a point that belongs to two (or possibly more in the case of non-proper representations) distinct curves in R. We use $\mathrm{In}(R)$ to denote the set of all intersection points in the string representation R.

Definition 25 (Precisely-k-string graphs)**.** A *precisely-k-string representation* is a proper representation R such that every pair of curves has either precisely k intersection points or precisely 0 intersection points.

A graph is a *precisely-k-string graph* if it has a precisely-k-string representation.

Theorem 30 (Euler's formula for k-string graphs)**.** *Let G be a connected k-string graph with a k-string representation R and let f be the number of faces of R. For a vertex $v \in V(G)$, let p_v be the number of intersection points on $R(v)$ and let $p = \frac{1}{2}\sum_{v \in V(G)} p_v$ be the number of all intersection points. Then,*

$$|V(G)| - k|E(G)| + f \leq 2.$$

Moreover, if G is a connected graph with a precisely-k-string representation R, the equality holds.

Proof. The argument is similar to Euler's formula for 1-string graphs, with some exceptions in the second paragraph. We have p intersection points and we can now observe that $|V(H(R))| = p$ and $|E(H(R))| = \sum_{v \in V(G)}(p_v - 1) = 2p - |V(G)|$. Again, we use Euler's formula for the planar graph $H(R)$: $2 = |V(H(R))| - |E(H(R))| + f = p - 2p + |V(G)| + f = |V(G)| - p + f \geq |V(G)| - k|E(G)| + f$, where the inequality follows from the fact that $p \leq k|E(G)|$, as each edge corresponds to a pair of strings with nonempty intersection, and every such pair interects in at most k points.

It is also clear that for a precisely-k-string representation, we have $p = k|E(G)|$, which yields equality. ⊞

We remark that in this case, the requirement for the representation to be proper is indeed necessary for the definition to be nontrivial. If we did not require the representation to be proper and instead only expected a finite number of intersection points and finitely many bends, we could then observe that the class of precisely-k-string graphs would be the same as the class of k-string graphs (with the same requirements on the representation, i.e., only finitely many intersection points and finitely many bends). Given a k-string graph and a pair of intersecting curves with less than k intersection points, we could create a representation with the two curves intersecting in precisely k intersection points by taking an intersection point of the two curves, and in a particularly small ε-neighborhood of the intersection point, add the remaining intersection points in such a way that the curves touch and do not cross.

A natural question also arises: what are the inclusions between the classes? Is there some hierarchy?

It turns out that a hierarchy of sorts is indeed present, however, it is somewhat complicated. We first start by showing that if we can represent a graph as a precisely-k-string, then we can represent it with any larger number of intersections of the same parity. After that, we show that for any precisely-k-string with k odd, we can find a precisely-$4k$-string representation, and therefore, it has a precisely-ℓ-string representation for any $\ell \geq 4k$.

However, we also show that not every precisely-k-string graph is a precisely-$(k+1)$-string graph which is quite unusual, as in most of hierarchies (k-string graphs, k-SEG graphs, B_k-VPG graphs), the sets are also ordered by inclusion. In

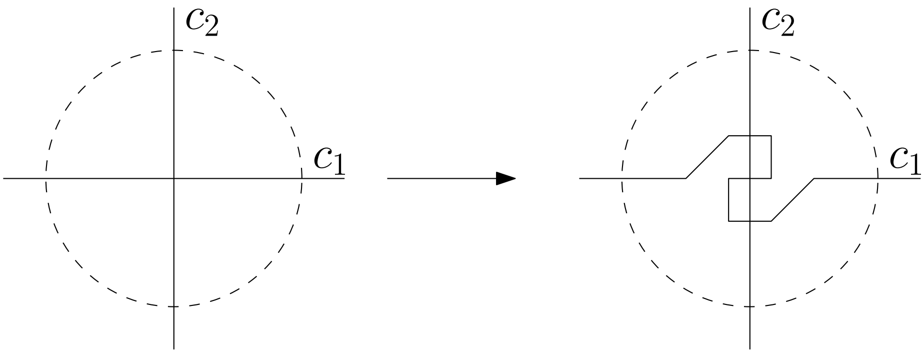

Figure 3.1: An example of adding two intersection points

our case, this only holds for steps of even length as mentioned before. Of course, we also show the more usual non-inclusion that there exists a precisely-$(k+1)$-string graphs that is not a precisely-k-string graph, which implies that the classes of precisely-k-string graphs and precisely-$(k+1)$-string graphs are incomparable with respect to inclusion.

3.2 Hierarchy inclusions

Proposition 31. *For all k, we have precisely-k-string $\subseteq$ precisely-$(k+2)$-string.*

Proof. We simply build a precisely-$(k+2)$-string representation from a precisely-k-string representation.

For every pair of intersecting curves c_1, c_2, we choose a single intersection point P of the two curves. As we assume the representations to be proper, there exists an $\varepsilon > 0$ such that the ε-neighborhood of P contains only parts of c_1, c_2 with P as their only intersection point and no other curve. We change the representation in the ε-neighborhood of P so that the curves intersect three times – c_2 will remain the same, while we change c_1 to cross c_2 three times as in Figure 3.1. ⊞

Proposition 32. *For all k, we have precisely-k-string $\subseteq$ precisely-$4k$-string.*

Proof. Given a precisely-k-string representation of G, we construct a precisely-$4k$-string representation of G. Intuitively, we "double" each string, which quadruples the number of intersections.

As the representation is proper, for each c in the representation, there exists an $\varepsilon_c > 0$ such that in the ε_c-neighborhood of c, no two other curves intersect, and if another curve in the representation enters the neighborhood, then it intersects c precisely once and then leaves the neighborhood. We then take $\varepsilon := \min_{c \in R} \varepsilon_c$.

We extend each curve c as follows: we take the curve c, draw another curve c' parallel to c inside the $\varepsilon/2$-neighborhood and join one of the two pairs of $\varepsilon/2$-close endpoints together by a single line segment, and an example is shown in Figure 3.2. It is clear that given two curves c, d intersecting in the point P, there are all four intersections in the ε-neighborhood of P: the intersections between c and d, c' and d and d' and c exist immediately by the construction. The intersection between c' and d' exists as the curves intersect the boundary of

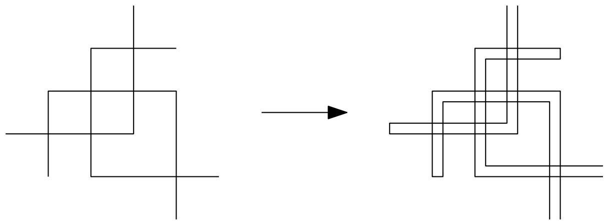

Figure 3.2: An example of multiplying the number of intersection points by four

the ε-neighborhood in the cyclic order $c, c', d, d', c', c, d', d$, and therefore c' and d' must intersect as well.

No more intersections are created anywhere else, which ensures that we indeed have a precisely-$4k$-string representation.

⊞

Proposition 33. *For all k, and for any bipartite graph G, if $G \in$ precisely-k-string, then $G \in$ precisely-$2k$-string.*

Proof. We repeat the construction from the proof of the previous theorem, however, we only apply the "doubling" operation on the curves representing the vertices of a single partition of G. This ensures that there are precisely $2k$ intersection points between two curves with a nonempty intersection. ⊞

We remark that the previous proposition does not follow from Proposition 31 as it may happen that k is odd, in which case the Proposition 31 does not imply the inclusion.

Theorem 34. *For all k, k-string $\subseteq$ precisely-$4k$-string.*

Proof. We start by creating a $4k$-string representation by "doubling" the string as in the proof of Proposition 31. In such a representation, we know by the construction that every pair of strings that has a nonempty intersection intersects in 4ℓ points for some $\ell \in \mathbb{N}$. In particular, the number of intersections is always even.

Therefore, we may continue with the argument as in Proposition 31 and for every pair of curves that has less than $4k$ intersection points, we add 2 intersection points until it has precisely $4k$ intersection points. We note that we cannot skip $4k$ intersections as we start with an even number of intersections, and by adding two intersection points, the parity of the number of intersection points remains unchanged. ⊞

Naturally, we may ask whether k-string is contained in precisely-ℓ-string for some $\ell < 4k$. This remains open; however, in the next section, we show that k-string is not contained in precisely-$(k+1)$-string.

3.3 Hierarchy non-inclusions

This section focuses on proving the non-inclusions precisely-k-string $\not\subseteq$ precisely-$(k+1)$-string and precisely-$(k+1)$-string $\not\subseteq$ precisely-k-string. We do this by employing the Noodle-Forcing Lemma of Chaplick et al. [10], which we use to reduce the task of finding a precisely-$(k+1)$-string to finding such representation by only extending the strings from their endpoints with the additional property that all added intersection points by essentially "doubling" consecutive intersection points at the beginning and the end of each string. The second non-inclusion then follows immediately from the Noodle-Forcing Lemma.

3.3.1 The Noodle-Forcing Lemma

Before we prove the theorems, we have to investigate the Noodle-Forcing Lemma and its proof. In particular, we aim to utilize the properties of the original construction while making the whole construction precisely-k-string. We begin by stating the lemma with the necessary definitions.

Definition 26 (Order-preserving mapping). Given a proper representation R, and a not necessarily proper representation R', with both representing the graph G, and a mapping $\varphi\colon \mathrm{In}(R) \to \mathrm{In}(R')$, we say that φ is *order-preserving* if it is injective and for every $v \in V$, if $p_1, \dots, p_k$ are all distinct intersection points on $R(v)$, then $\varphi(p_1), \dots, \varphi(p_k)$ all belong to $R'(v)$ and they appear on $R'(v)$ in the same relative order as the points $p_1, \dots, p_k$ on $R(v)$.

Lemma 35 (Noodle-Forcing Lemma, Chaplick et al. [10]). *Let $G = (V, E)$ be a graph with a proper representation $R = \{R_v : v \in V\}$. Then there exists a graph $G' = (V', E')$ containing G as an induced subgraph, which has a proper representation $R' = \{R'(v) : v \in V'\}$ such that $R(v) = R'(v)$ for all $v \in V$ and $R'(w)$ is a vertical or a horizontal segment for $w \in V' \setminus V$.*

Moreover, for any $\varepsilon > 0$ any (not necessarily proper) representation of G' can be transformed by a homeomorphism of the plane and a circular inversion into a representation $R^\varepsilon = \{R^\varepsilon(v) : v \in V'\}$ with these properties:

1. *for every vertex $v \in V$, the curve $R^\varepsilon(v)$ is contained in the $\varepsilon-$neighborhood of $R(v)$ and $R(v)$ is contained in the $\varepsilon-$neighborhood of $R^\varepsilon(v)$.*
2. *there is an order-preserving mapping $\phi : In(R) \to In(R^\varepsilon)$ with the additional property that for every $p \in In(R)$, the point $\phi(p)$ coincides with the point p.*

We briefly sketch the construction, which works in two steps: in the first step, we overlay the proper representation R of a graph F by a plane grid graph H with particular properties, and in the second step, we construct a grid intersection representation based on H.

The first step is performed as follows. We start by defining *special* points of R – these are the endpoints of the curves, bend points of the curves, and intersection points of the curves. We then construct the plane grid graph H that overlays the representation with the following properties:

(P1) The edges of H are drawn as vertical and horizontal segments, and every internal face of H is a rectangle. Moreover, the outer face of H does not intersect any curve of R.

(P2) No curve of R passes through a vertex of H and no edge of H passes through a special point of R.

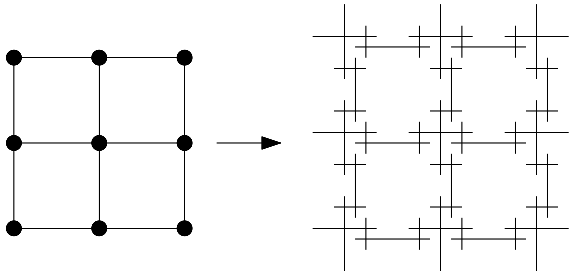

Figure 3.3: The representation of cells in Noodle-Forcing Lemma

(P3) Every face of H contains at most one special point of R and no two faces containing a special point are adjacent.

(P4) Every edge of H has at most a single intersection with the curves of R.

(P5) Every face of H intersects at most two curves of R, and if a face f intersects exactly two curves of R, then the two curves intersect inside the face f.

(P6) Every curve of R intersects the boundary of a face of H at most twice.

The first three properties can be obtained by taking a grid that is large enough and fine enough. The remaining three properties are then ensured by suitably splitting the faces of H.

We proceed with the second step. Given the plane graph H, we build a representation R' of the graph G' as follows. Starting with the vertices of H, each $v \in V(H)$ is converted into two vertices $S_1(v), S_2(v)$ which form an edge together. Each edge $\{u, v\} = e \in E(H)$ that is incident with v is converted into three vertices $S(v, e), S(u, e), S(e)$ with edges $\{S(v, e), S(e)\}, \{S(u, e), S(e)\}$. If an edge e that is incident with a vertex v is drawn as a horizontal segment, we add an edge between $S(v, e)$ and $S_1(v)$ and if e is drawn as a vertical segment, the edge is added between the vertices $S(v, e)$ and $S_2(v)$. We represent this naturally with a grid intersection representation as shown in Figure 3.3.

To finish the construction, we add the vertices of G as in the representation R with edges given by the intersections. An example of such reduction is shown in Figure 3.4.

3.3.2 Proofs of the non-inclusions

Our first goal is to construct a precisely-k-string graph that does not have a precisely-$(k + 1)$-string representation. We do this in two steps. First, we show that given a precisely-k-string representation R of a graph G, we create a graph $G' = G'(R)$ that has a precisely-$(k + 1)$-string representation if and only if the representation R can be extended into a precisely-$(k+1)$-string representation by "doubling" consecutive intersection points at the beginning and the end of each string by extending the string from its endpoints. In the second step, we find a precisely-k-string graph with a representation that cannot be extended in such a way. We first formalize the concept of extending the representation.

Definition 27 (Suitable extension)**.** We say that arepresentation R' of a graph G *suitably extends* a representation R of a graph G if

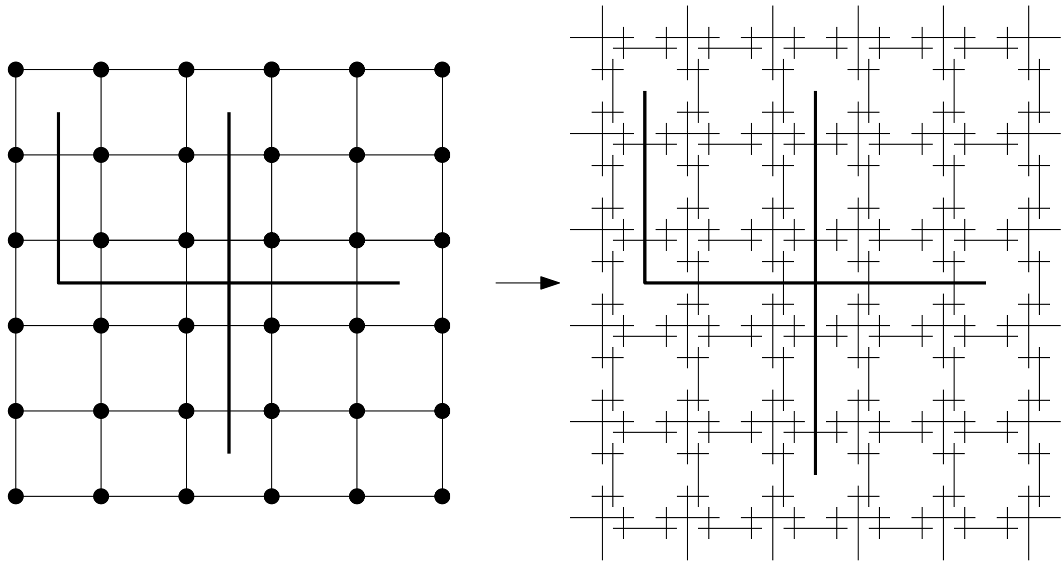

Figure 3.4: An example of the reduction in the proof of Noodle-Forcing Lemma

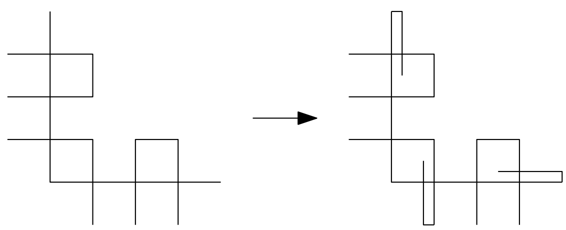

Figure 3.5: An example of a suitable extension of a precisely-2-string representation into a precisely-3-string representation

1. $\forall v \in V(G), R(v) \subseteq R'(v)$, and
2. if $p_1, \dots, p_\ell$ are all distinct intersection points in the order they appear on $R'(v)$ and $p_a, \dots, p_b$, $1 \le a \le b \le \ell$ are all intersection points that appear on $R(v)$, then $\forall i \in [a-1]$, the points p_i, p_{2a-i} intersect the same curve $R(u)$ and there is no other intersection point between p_i and p_{2a-i} on $R(u)$, and $\forall i \in [\ell - b]$ the points p_{b+i}, p_{b+1-i} intersect the same curve $R(u)$ and there is no other intersection point between p_{b+i} and p_{b+1-i} on $R(u)$.

An example of a suitable extension is shown in Figure 3.5.

Lemma 36. *For all $k \ge 1$, given a precisely-k-string representation R of a graph G, the graph G' constructed in the Noodle-Forcing Lemma has a precisely-k-string representation as well.*

Proof. Let us take the representation R'' as in the proof of the Noodle-Forcing Lemma and then we add the missing required intersections.

We now have to take care of the intersections of the newly added strings and the intersections between the original strings and the newly added strings. We start by resolving the issues with pairs of the newly added strings from the Noodle-Forcing Lemma. We added three types of edges: the edges $\{S_1(v), S_2(v)\}$, $\{S(v, e), S_i(v)\}$, and $\{S(v), S(v, e)\}$. In the representation R'', any such edge corresponds to a single intersection point, therefore, we have to add k more. In the

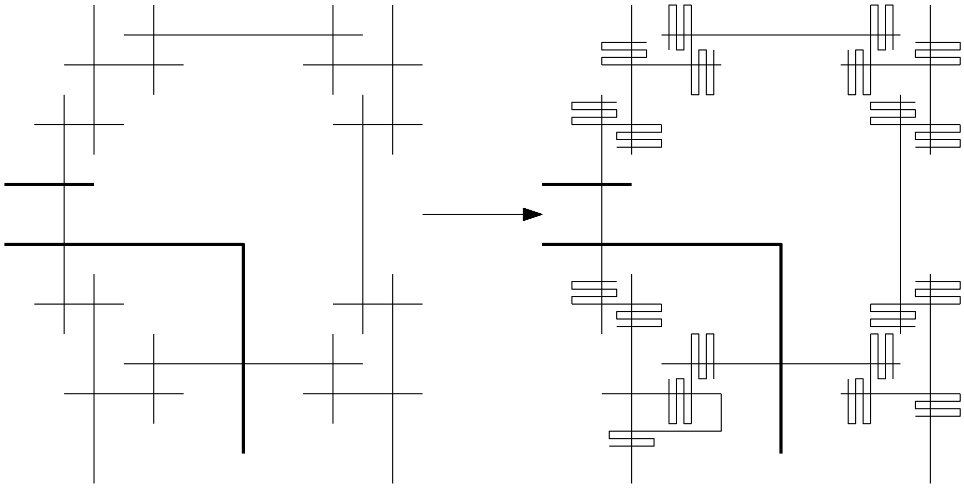

Figure 3.6: The first step of extension of intersection points

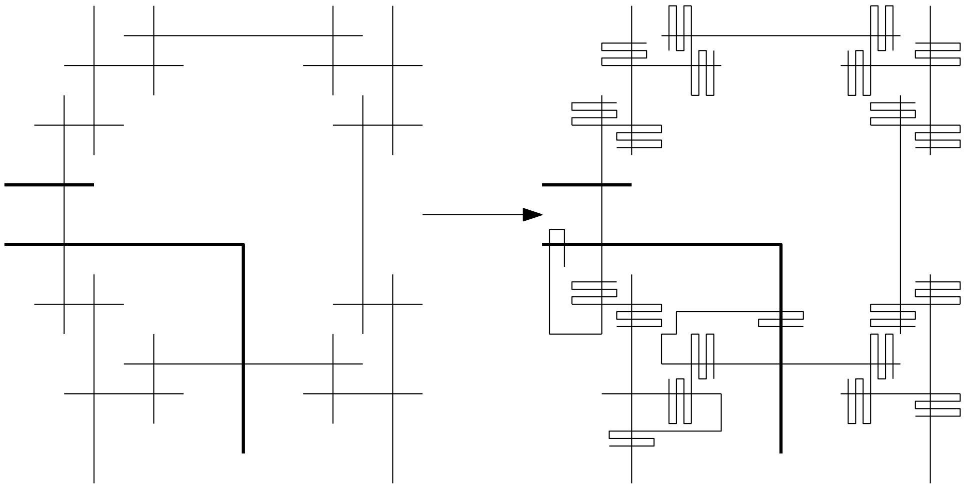

Figure 3.7: The final step of extension of intersection points

case of edges $\{S(v,e), S_i(v)\}$ and $\{S(v), S(v,e)\}$, we extend the connector $S(v,e)$, so that on the extension on one endpoint, we add the intersections with $S(v)$ and on the extension of the other endpoint, we add the required intersections with $S_i(v)$. The edges $\{S_1(v), S_2(v)\}$ are resolved similarly, we extend one endpoint of $S_1(v)$ to add the required intersections. We show this graphically in Figure 3.6.

Finally, we take care of the intersections between the newly added strings and the original strings. The only permitted edges are of the form $\{S(e), v\}$ for $v \in V(G)$. Nevertheless, we may still require more intersection points, which we accomplish by extending the strings representing the vertices $S(e)$ from one of the endpoints. The result is shown in Figure 3.7.

$\boxplus$

Lemma 37. *For all $k \geq 1$, given a precisely-k-string representation R of a graph G, there exists a graph $G' = G(R)$ that has a precisely-$(k+1)$-string representation if and only if the representation R can be suitably extended into a precisely-$(k+1)$-string representation R' of the graph G.*

Proof. Given a precisely-k-string representation R of a graph G, we apply the Noodle-Forcing Lemma to the representation R and we obtain a graph G' with its representation R'' as described in the previous section. We claim that G' satisfies the properties stated in Lemma 37.

We first prove the converse. Let us have a precisely-$(k + 1)$-string representation R' of G that suitably extends R. We can then simply take the graph G' with its representation R'' and extend each string from its endpoints so that it has the additional intersections from the suitable extension by drawing the new parts ε-close to the original string with ε small enough so that the ε-neighborhood of $R''(v)$ contains only parts of the curves that $R''(v)$ intersects and if a curve c enters the ε-neighborhood of $R''(v)$, then it crosses $R''(v)$ a single time before leaving the neighborhood. This forms the representation $\tilde{R}$. Moreover, the construction ensures that all pairs of vertices that were present in G intersect in either 0 or $k + 1$ points in $\tilde{R}$. We then use the same approach as in the proof of Lemma 36.

We also remark that in the case of the intersections between the newly added strings and the original strings, there may be more than a single intersection between the representations of $S(e)$ and v. In such case, we can have either one or two intersections, depending on whether the representation of vertex v has been extended that far. In particular, we note that it cannot happen that we would have more intersections than required as the extension adds at most a single intersection with each already intersected string. This finishes the proof of the converse, and now we move to the forward implication.

We are now given a precisely-$(k + 1)$-string representation R' of G' and we want to show that it yields a representation R'' that suitably extends R. By the Noodle-Forcing Lemma, we can transform the proper representation R' into a representation R^ε of G' with properties as in the statement of the lemma. Moreover, R^ε is also a precisely-$(k+1)$-string representation as homeomorphisms and circular inversions are bijections, and hence the intersections are all preserved as crossings and no new intersections appear.

As $R^\varepsilon(v)$ is ε-close to $R(v)$, we note that $R^\varepsilon(v) \subseteq N_\varepsilon(R(v)) = \{x\colon \exists y \in R(v) : \mathrm{dist}(x, y) < \varepsilon\}$. Using the same terminology as Chaplick et al. [10], we focus on the *zones* of the representation and we call $N_\varepsilon(R(v))$ the *noodles*. Let two curves $R(u), R(v)$ have k mutual crossing points $p_1, \dots, p_k$. The proof of the Noodle-Forcing Lemma requires ε to be small enough so that $N_\varepsilon(R(v)) \cap N_\varepsilon(R(u))$ has k connected components $Z_1, \dots, Z_k$, all of which are parallelograms, and all the connected components are disjoint. We call these connected components the *zones* with each Z_i containing the point p_i. We also add the requirement for ε to be small enough so that the distance between any intersection point in R and any endpoint of a curve in R is strictly larger than $\varepsilon > 0$. This is possible due to the fact that R is proper.

We note that as R^ε is a precisely-$(k + 1)$-string representation, there is only a single zone with two intersection points for each pair of vertices u, v forming an edge, as there is one intersection point in each of the zones by the Noodle-Forcing Lemma. Moreover, as we chose ε small enough so that for any vertex $u \in V(G)$ the distance between any intersection point of $R(u)$ and the endpoints is greater than ε, by the ε-closeness of $R(u)$ and $R^\varepsilon(u)$, we have that in each of the zones, the curves must intersect both of their respective opposite sides of the boundary of the zone so that ε-closeness can be preserved. We show that the only possible way of adding a single intersection point into a zone is by having one of

$R^\varepsilon(u), R^\varepsilon(v)$ cross the zone twice. It may happen that $R^\varepsilon(u)$ intersects Z_i with multiple subcurves, and we focus on the different variants of these subcurves.

There are only two possibilities of the positions of subcurves in the zone Z_i based on the endpoints: either both endpoints of the subcurve lie on the same side of the parallelogram or they lie on the opposite sides. If there was a subcurve that would have endpoints on two neighboring sides, this would be a contradiction with the fact that for any vertex w, $R^\varepsilon(w) \subseteq N_\varepsilon(R(w))$ as $N_\varepsilon(R(w))$ is an open set and one of the sides would lie on the boundary of the noodle $N_\varepsilon(R(w))$, which is not part of the set.

We first focus on the subcurves in Z_i that have both endpoints on the same side of the parallelogram. In this case, the subcurve $\gamma \subseteq R^\varepsilon(u)$ splits Z_i into two regions with one region bounded by the curve γ and the line segment between the two endpoints of γ lying on the boundary of Z_i, and therefore any curve $\delta \subseteq R^\varepsilon(v)$ that would cross γ must cross it at least twice (or, more generally, an even number of times) as δ cannot leave the region through the line segment. (Here, we extensively use the fact that we do not permit two curves to touch without crossing.) Therefore, the subcurves with both endpoints of the subcurve on the same side of the zone cannot add just a single intersection point.

This means that we cannot add just a single intersection by any pair of subcurves such that at least one of them has both endpoints on the same side of the zone. We therefore now focus on the case of two subcurves $\gamma \subseteq R^\varepsilon(u)$, $\delta \subseteq R^\varepsilon(v)$ with the endpoints of each subcurve on the opposite sides of the zone. Again, γ divides Z_i into two regions with one endpoint of δ in each of the regions. Therefore, as δ cannot leave the two regions, it must cross γ odd number of times, and therefore we cannot get precisely two intersection points between δ and γ.

It immediately follows that the only way how to get two intersection points in a single zone is by having exactly one of the two curves $R^\varepsilon(u), R^\varepsilon(v)$ have two subcurves that have their endpoints on the opposite sides of the zone. For brevity, we say that the subcurve crosses the zone if its endpoints lie on the opposite sides of the zone.

We now build the representation $R'' = \{R''(v) = R^\varepsilon(v) \colon v \in V(G)\}$ of G by restricting R^ε.

We first show that the second property of a suitable extension holds for R''. We fix a vertex v and we choose an orientation of $R(v)$. Let us order the zones $Z_1, \dots, Z_\ell$ by the order of their corresponding intersection points on $R(v)$ with respect to the chosen orientation. We now show that it cannot happen that there exists a zone Z_j such that $R''(v)$ has two crossing subcurves through the zone Z_j, but there exist zones $Z_i, Z_k : i < j < k$ such that $R''(v)$ has only a single crossing subcurve through the zones Z_i, Z_k. For contradiction, we assume such zones exist as is shown in Figure 3.8 and we consider the disjoint sets A, B, C, D as depicted there. Each of the four sets has an odd number of crossings with $R''(v)$ on the boundary, and therefore, there must be an endpoint in each of the four sets. However, curves only have two endpoints, and this yields a contradiction. Therefore, the zones with two crossing subcurves must form two continuous intervals $Z_1, \dots, Z_m$ for some m and $Z_k, \dots, Z_\ell$ for some k.

To get the first property, we can just simply use a single homeomorphism to deform the part $R''(v)$ containing the original intersection points into $R(v)$ as we now can only focus on intersections with other vertices of the original graph G

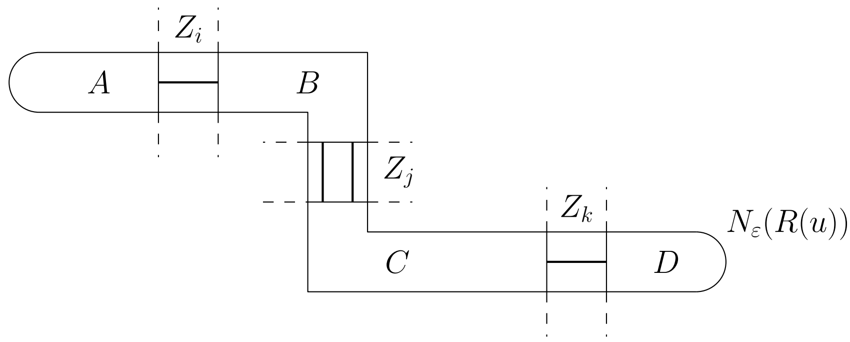

Figure 3.8: The impossible situation with zones in proof of Lemma 37

in the zones of the noodle $N_\varepsilon(R(v))$. Therefore, the representation R'' indeed suitably extends R. ⊞

Lemma 38. *For all $k \geq 1$, there exists a graph G_k with its precisely-k-string representation R_k such that the representation cannot be suitably extended into a precisely-$(k+1)$-string representation.*

Proof. We start with the case $k = 1$, where the graph is $G_1 = K_8$, the complete graph on eight vertices, with its representation R_1 as in Figure 3.9. We want to show that there is no suitable extension of the representation and for contradiction, we assume that such extension exists. We will consider the added intersections of vertices c_1, c_2, d_1, d_2.

First, we note that only one of c_1, c_2 and one of d_1, d_2 can be extended from the top endpoint farther than their intersection as otherwise, we would get more than two intersection points for the two vertices (let c_x, d_y be the vertices that are extended from the top endpoint farther than their intersection with c_{3-x}, d_{3-y} respectively). This also implies that at least one of the four intersections between c_i, d_j cannot be added by the top extension – in particular, the intersection between c_{3-x}, d_{3-y}. We note that some configurations of these extensions cannot be extended from the bottom (e.g., if c_2 is extended as far down as possible and d_1 is extended to its intersection with c_1, then the intersection point of c_1, d_2 cannot be reached from the bottom), however, such cases cannot be the extensions either way.

We can also use the argument to see that there is at least one intersection point between a_i and b_j that is not covered by the extensions starting from the top. Therefore, at least one of a_i, b_i must be extended from the bottom to obtain the uncovered intersection and the same is true for at least one c_j, d_j. However, each pair with one vertex a_i or b_i and the other vertex c_j or d_j has an intersection in the bottom half of the representation, and hence it cannot happen that both of them can be extended up to their so far uncovered intersection. Therefore, the representation cannot be suitably extended.

Next, we continue with the case $k \geq 2$. In this case, the graph is $G_k = K_4$, the complete graph on four vertices, with the representation R_k as in Figure 3.10. We start by noting that at most one of the strings a_1 and a_2, or b_1 and b_2 respectively, can be extended as both the leftmost and the rightmost intersection point are between a_1, a_2, or b_1, b_2 respectively. Therefore, there are an a_i and a b_j that cannot be extended and therefore, the number of intersection points between a_i and b_j remains at k, showing that we cannot suitably extend the representation.

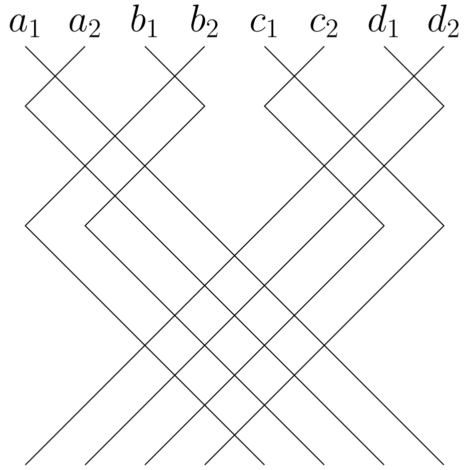

Figure 3.9: The graph G_1 with representation R_1

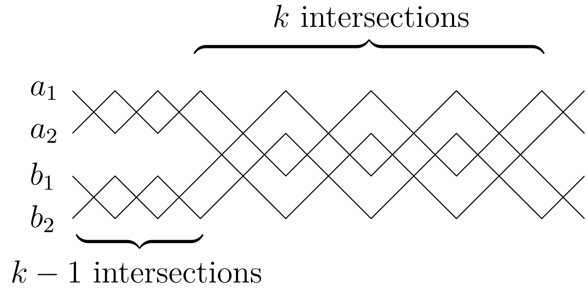

Figure 3.10: The graph G_k with representation R_k for $k \geq 2$

⊞

Theorem 39. *For all $k \geq 1$, precisely-k-string $\not\subseteq$ precisely-$(k+1)$-string.*

Proof. This follows immediately: from Lemma 38, there is a graph G_k with its representation R_k that cannot be suitably extended, and therefore, the graph G'_k from Lemma 37 has a precisely-k-string representation, but it does not have a precisely-$(k+1)$-string representation, proving the theorem. ⊞

Corollary 40. *For all $k \geq 1$, k-string $\not\subseteq$ precisely-$(k+1)$-string.*

Proof. Any precisely-k-string graph is a k-string graph and hence there exists a k-string graph that is not a precisely-$(k+1)$-string graph. ⊞

Our second goal is to show that the opposite inclusion does not hold, that is, there exists a precisely-$(k+1)$-string graph that is not a precisely-k-string graph.

Theorem 41. *For all $k \geq 1$, precisely-$(k+1)$-string $\not\subseteq$ precisely-k-string.*

Proof. We construct a representation R_{k+1} of a graph $G = K_2$, the complete graph on two vertices. In particular, the construction is the same as in the proof of B_k-VPG$\subsetneq$$\mathrm{B}_{k+1}$-VPG by Chaplick et al. [10]. As shown in Figure 3.11, the represention is precisely-$(k+1)$-string, and we apply the Noodle-Forcing Lemma on the representation to obtain a graph G' with representation R'_{k+1}. (Using terminology by Chaplick et al. [10], we *grill the sausage*.) By Lemma 36, the graph G' has a precisely-$(k+1)$-string representation.

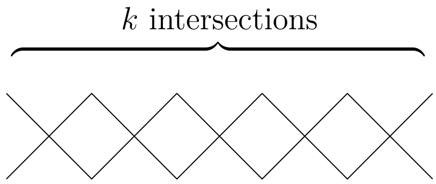

Figure 3.11: The sausage construction of representation R_k

We now claim that the graph G' has no precisely-k-string representation. For contradiction, let R'' be a precisely-k-string representation of G'. By the Noodle-Forcing Lemma, we can use a homeomorphism of the plane and circular inversion to transform R'' into a representation R^ε such that there exists an order-preserving mapping $\phi : In(R_{k+1}) \to In(R^\varepsilon)$. Moreover, as we observed in the proof of Lemma 37, any R^ε obtained by transforming a precisely-k-string representation must be precisely-k-string as well.

Let us consider $R_{k+1}(u), R_{k+1}(v)$ with u, v the two distinct vertices of G. However, $R_{k+1}(u)$ has $k+1$ intersection points with $R_{k+1}(v)$, and as φ is injective by definition, there must be at least $k+1$ intersection points between $R^\varepsilon(u)$ and $R^\varepsilon(v)$, which is a contradiction. ⊞

All of this yields a rather surprising structure in the hierarchy, as shown in Figure 3.12. We have two chains of classes for k even and odd, respectively, and we have a connection showing that for every odd k, we can find a representation with an even number of $4k$ intersections per pair of curves. However, the other direction is not clear at all: can we find a representation with $2\ell + 1$ intersections per pair of curves for any precisely-$2k$-string?

3.4 Open problems

We now mention two problems that remain open. The first problem consists of finding representations with an odd number of intersections while having an even number of intersections. The other direction follows from Proposition 32, but it is not clear if we can change the parity of the number of intersections of two curves from even to odd.

Problem. Given a $k \in \mathbb{N}$, does there exist some $\ell \in \mathbb{N}$ such that precisely-$2k$-string $\subseteq$ precisely-$(2\ell + 1)$-string?

The second problem consists of minimizing the number of intersections ℓ such that k-string graph can be represented as a precisely-ℓ-string graph. We are able to show that $\ell \leq 4k$ and that $\ell > k + 1$. The main issue we run into is that some pairs of curves may have an odd number of intersections and some pairs may intersect in an even number of points. Resolving this with a less intersection inflating procedure could quickly strengthen the bound.

Problem. What is the least ℓ such that k-string $\subseteq$ precisely-ℓ-string?

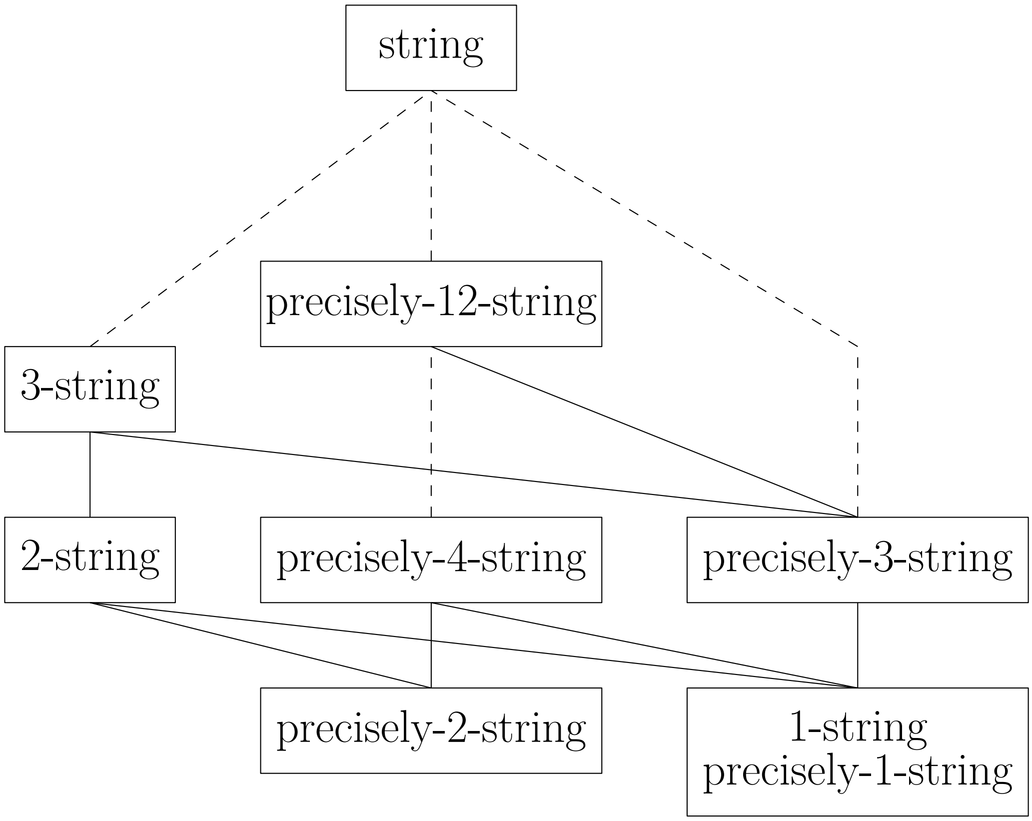

Figure 3.12: The diagram of inclusions of the investigated classes

Conclusion

In the thesis, we studied different types of hierarchies of intersection graphs. First, we considered the problem of finding a class of intersection graphs that contains all graphs with maximum degree at most d for any $d \in \mathbb{N}$. This led to the introduction of classes of pure-k-line, pure-k-tile, impure-k-line and impure-k-tile graphs.

We first established relations between these classes with k serving as a graph parameter, and we also observed that there are graphs with bounded pure-line parameter while the pure-tile parameter is unbounded. Next, we considered the relationship between the parameters and maximum degree or graph degeneracy. We showed that d-degeneracy of a graph G implies that G has a pure-d-tile representation and a graph G' with maximum degree at most $2d$ has an impure-d-tile representation. Using Ramsey theory, we were also able to show that there exist bipartite graphs with arbitrarily large impure-tile and impure-line parameters. We also compared the introduced parameters with other known graph parameters such as treewidth, pathwidth, or clique-width. In all three cases, we showed that there exist graphs with bounded pure-line parameter but unbounded treewidth, pathwidth and clique-width. On the other hand, bounded treewidth and pathwidth imply bounded pure-line parameter. The case of clique-width remains open.

When considering the complexity of recognizing graphs with parameter $\leq k$ for any of the four parameters, we proved a version of Euler's formula for 1-string graphs. This motivated the definition of the class of precisely-k-string graphs, which we studied in Chapter 3. We showed inclusions of precisely-k-string graphs into precisely-$(k+2)$-string graphs and precisely-$4k$-string graphs. We also gave examples of graphs that are precisely-k-string graphs (and hence also k-string graphs) but are not precisely-$(k+1)$-string graphs.

In the thesis, we introduced many open problems, two of which we point out here. The first problem considers the complexity of deciding whether a general graph has its pure-tile parameter $\leq k$. It is easy to observe that the decision problem is in NP, but it is not clear whether it is NP-complete or not.

In the second problem, we consider k-string graphs and their extensions into precisely-ℓ-string graphs. We were able to show that every k-string graph is a precisely-$4k$-string graph. Is this the best possible, or can we find a smaller ℓ such that every k-string graph is a precisely-ℓ-string graph?

Bibliography

[1] Andrei Asinowski, Elad Cohen, Martin Charles Golumbic, Vincent Limouzy, Marina Lipshteyn, and Michal Stern. Vertex intersection graphs of paths on a grid. *J. Graph Algorithms Appl.*, 16(2):129–150, 2012.

[2] Martin Charles Golumbic, Marina Lipshteyn, and Michal Stern. Edge intersection graphs of single bend paths on a grid. *Networks*, 54(3):130–138, 2009.

[3] William T. Trotter Jr. and Frank Harary. On double and multiple interval graphs. *Journal of Graph Theory*, 3(3):205–211, 1979.

[4] Jerrold R. Griggs and Douglas B. West. Extremal values of the interval number of a graph. *SIAM Journal on Algebraic Discrete Methods*, 1(1):1–7, 1980.

[5] András Gyárfás and Douglas West. Multitrack interval graphs. *Congr. Numerantium*, 109:109–116, 1995.

[6] Reuven Bar-Yehuda, Magnús M Halldórsson, Joseph Naor, Hadas Shachnai, and Irina Shapira. Scheduling split intervals. *SIAM Journal on Computing*, 36(1):1–15, 2006.

[7] Liliana Alcón, Márcia R. Cerioli, Celina M.H. de Figueiredo, Marisa Gutierrez, and João Meidanis. Tree loop graphs. *Discrete Applied Mathematics*, 155(6):686–694, 2007. Computational Molecular Biology Series, Issue V.

[8] Vineet Bafna, Babu Narayanan, and R. Ravi. Nonoverlapping local alignments (weighted independent sets of axis-parallel rectangles). *Discrete Applied Mathematics*, 71(1):41–53, 1996.

[9] Jan Kratochvíl and Jiří Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62(2):289–315, 1994.

[10] Steven Chaplick, Vít Jelínek, Jan Kratochvíl, and Tomáš 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.

[11] Jan Kratochvíl and Jiří Matoušek. NP-hardness results for intersection graphs. *Commentationes Mathematicae Universitatis Carolinae*, 030(4):761–773, 1989.

[12] Douglas B. West and David B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discrete Applied Mathematics*, 8(3):295–305, 1984.

[13] Minghui Jiang. Recognizing d-interval graphs and d-track interval graphs. *Algorithmica*, 66(3):541–563, 2013.

[14] Daniel Heldt, Kolja Knauer, and Torsten Ueckerdt. Edge-intersection graphs of grid paths: The bend-number. *Discrete Applied Mathematics*, 167:144–162, 2014.

[15] Martin Pergel and Paweł Rzążewski. On edge intersection graphs of paths with 2 bends. In *Graph-Theoretic Concepts in Computer Science*, pages 207–219. Springer, 2016.

[16] Dror Epstein, Martin Charles Golumbic, Abhiruk Lahiri, and Gila Morgenstern. Hardness and approximation for l-EPG and B1-EPG graphs. *Discrete Applied Mathematics*, 281:224–228, 2020. LAGOS'17: IX Latin and American Algorithms, Graphs and Optimization Symposium, C.I.R.M., Marseille, France - 2017.

[17] Kellogg S. Booth and George 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.

[18] Michel Habib, Ross McConnell, Christophe Paul, and Laurent Viennot. Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234(1):59–84, 2000.

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

[20] Marcus Schaefer, Eric Sedgwick, and Daniel Štefankovič. Recognizing string graphs in NP. *Journal of Computer and System Sciences*, 67(2):365–380, 2003. Special Issue on STOC 2002.

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

[22] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 460–467, New York, NY, USA, 1988. Association for Computing Machinery.

[23] Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. *CoRR*, abs/1406.2636, 2014.

[24] Andrei Asinowski and Andrew Suk. Edge intersection graphs of systems of paths on a grid with a bounded number of bends. *Discrete Applied Mathematics*, 157(14):3174–3180, 2009.

[25] Petr Chmel. Algorithmic aspects of intersection representations. Bachelor's thesis, Charles University, Faculty of Mathematics and Physics, 2020.

[26] Dibyayan Chakraborty and Kshitij Gajjar. Finding geometric representations of apex graphs is NP-hard. In *International Conference and Workshops on Algorithms and Computation*, pages 161–174. Springer, 2022.

[27] Kathie Cameron, Steven Chaplick, and Chính T. Hoàng. Edge intersection graphs of L-shaped paths in grids. *Discrete Applied Mathematics*, 210:185–194, 2016.

[28] Antoon W.J. Kolen, Jan Karel Lenstra, Christos H. Papadimitriou, and Frits C.R. Spieksma. Interval scheduling: A survey. *Naval Research Logistics (NRL)*, 54(5):530–543, 2007.

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

[30] Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry*, 9(1):3–24, 1998. Special Issue on Geometric Representations of Graphs.

[31] Ross J Kang and Tobias Müller. Sphere and dot product representations of graphs. *Discrete & Computational Geometry*, 47(3):548–568, 2012.

[32] Irina Mustaţă and Martin Pergel. What makes the recognition problem hard for classes related to segment and string graphs? *CoRR*, abs/2201.08498, 2022.

[33] Jan Kratochvíl and Martin Pergel. Geometric intersection graphs: do short cycles help? In *International Computing and Combinatorics Conference*, pages 118–128. Springer, 2007.

[34] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM Journal on Discrete Mathematics*, 5(4):596–603, 1992.

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

[36] Paul Turán. On an extremal problem in graph theory. *Matematikai és Fizikai Lapok*, 48:436–452, 1941.

[37] Willem Mantel. Problem 28 (solution by H. Gouwentak, W. Mantel, J. Teixeira de Mattes, F. schuh and W. A. Wythoff). *Wiskundige Opgaven*, 10:60–61, 1907.

[38] F. P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, s2-30(1):264–286, 1930.

[39] Reinhard Diestel. *Graph Theory*. Springer, Heidelberg, 2017.

[40] Arkadiusz Pawlik, Jakub Kozik, Tomasz Krawczyk, Michał Lasoń, Piotr Micek, William T Trotter, and Bartosz Walczak. Triangle-free geometric intersection graphs with large chromatic number. *Discrete & Computational Geometry*, 50(3):714–726, 2013.

[41] Julius Petersen. Die theorie der regulären graphs. *Acta Mathematica*, 15(1):193–220, 1891.

[42] Louigi Addario-Berry (https://mathoverflow.net/users/3401/louigi-addario-berry). Proving that every graph is an induced subgraph of an r-regular graph. MathOverflow. URL: https://mathoverflow.net/q/52375 (version: 2011-01-18).

[43] Jaroslav Nešetřil and Vojtěch Rödl. Simple proof of the existence of restricted ramsey graphs by means of a partite construction. *Combinatorica*, 1(2):199–202, 1981.

[44] Paul Erdős, András Hajnal, and Richard Rado. Partition relations for cardinal numbers. *Acta Math. Acad. Sci. Hungar*, 16:93–196, 1965.

[45] János Pach, Hubert de Fraysseix, and Patrice Ossona de Mendez. Representation of planar graphs by segments. Technical report, North-Holland, 1994.

[46] Jiří Fiala. Graph minors, decompositions and algorithms. https://kam.mff.cuni.cz/~fiala/tw.pdf, 2022. Accessed: 2022-04-06.

[47] Rudolf Halin. S-functions for graphs. *Journal of geometry*, 8(1):171–186, 1976.

[48] Prosenjit Bose, Jonathan F. Buss, and Anna Lubiw. Pattern matching for permutations. *Information Processing Letters*, 65(5):277–283, 1998.

[49] Elad Cohen, Martin Charles Golumbic, and Bernard Ries. Characterizations of cographs as intersection graphs of paths on a grid. *Discrete Applied Mathematics*, 178:46–57, 2014.

[50] Vít Jelínek. The rank-width of the square grid. *Discrete Applied Mathematics*, 158(7):841–850, 2010. Third Workshop on Graph Classes, Optimization, and Width Parameters Eugene, Oregon, USA, October 2007.

[51] Sang il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.

[52] E.R Scheinerman and J Zito. On the size of hereditary classes of graphs. *Journal of Combinatorial Theory, Series B*, 61(1):16–39, 1994.

[53] Alice M Dean, Joan P Hutchinson, and Edward R Scheinerman. On the thickness and arboricity of a graph. *Journal of Combinatorial Theory, Series B*, 52(1):147–151, 1991.

[54] Anthony Mansfield. Determining the thickness of graphs is NP-hard. *Mathematical Proceedings of the Cambridge Philosophical Society*, 93(1):9–23, 1983.

[55] A.V. Kostochka and J. Nešetřil. Coloring relatives of intervals on the plane, I: Chromatic number versus girth. *European Journal of Combinatorics*, 19(1):103–110, 1998.

List of Figures

2.1 The operation of “purifying” the intervals 12
2.2 Building the L-graph representation of a complete k-partite graph 14

3.1 An example of adding two intersection points 28
3.2 An example of multiplying the number of intersection points by four 29
3.3 The representation of cells in Noodle-Forcing Lemma 31
3.4 An example of the reduction in the proof of Noodle-Forcing Lemma 32
3.5 An example of a suitable extension of a precisely-2-string representation into a precisely-3-string representation 32
3.6 The first step of extension of intersection points 33
3.7 The final step of extension of intersection points 33
3.8 The impossible situation with zones in proof of Lemma 37 36
3.9 The graph G_1 with representation R_1 37
3.10 The graph G_k with representation R_k for $k \geq 2$ 37
3.11 The sausage construction of representation R_k 38
3.12 The diagram of inclusions of the investigated classes 39