# $(k,p)$-Planar Graphs

## A Generalization of Planar Representations for Cluster Graphs

Timothy W. Randolph

Professor William J. Lenhart, Advisor

A thesis submitted in partial fulfillment of the requirements for the
Degree of Bachelor of Arts with Honors in Computer Science

Williams College
Williamstown, Massachusetts

April 28, 2018

# Contents

# List of Figures

# Abstract

A *cluster graph* consists of a graph $G = (V, E)$ and a partition of the vertex set $V$ into clusters $V_1, V_2, ...V_C$. We refer to an edge $(u, v) \in E$ as *intracluster* if it connects two vertices in the same cluster and *intercluster* otherwise. A *port drawing* of a cluster graph $G$ is a planar representation of $G$ in which every cluster $V_i$ is associated with a distinct *cluster region* $R_i$, each vertex $v \in V_i$ is associated with one or more *ports* on the perimeter of $R_i$, and each intercluster edge $(u, v) \in E$ is associated with a simple curve connecting a port of $u$ to a port of $v$. A port drawing is *planar* if no edge curves cross or enter cluster regions.

Previous cluster graph representations highlight structural features, minimize edge crossings, or attempt to do both. We introduce the $(k, p)$*-planar drawing*, a planar representation for cluster graphs that generalizes established cluster graph representations and allows for the flexible representation of cluster subgraphs within cluster regions. We say that a cluster graph $G$ is $(k, p)$*-planar* if $G$ admits a planar port drawing in which every vertex is associated with at most $p$ ports and no cluster contains more than $k$ vertices. This thesis relates the $(k, p)$-planar graphs to established graph classes, bounds the edge density of the $(k, p)$-planar graphs, provides hardness results for the problem of deciding whether or not a graph is $(k, p)$-planar, and considers extensions to the $(k, p)$-planar drawing schema that introduce intracluster representations.

# Acknowledgments

First, I am deeply grateful to my family for their constant support of all my projects, research and otherwise. Thanks to my friends, who sympathized and saw less of me during the weeks devoted to writing and editing my thesis. I would also like to thank the FOSS, inscrutable, unquenchable, and finally in print. I am grateful to Pamela Harris, who taught me to format and publish mathematics research and whose consistency, positivity and friendliness is an inspiration to me. I extend a special thanks to my collaborators Beppe Liotta, Emilio Di Giacomo and Alessandra Tappini at the University of Perugia for their friendliness, insight, and constructive criticism of my ideas. Thanks also to Duane Bailey, the second reader of this thesis.

Finally, I would like to thank my advisor, Bill Lenhart, for my introduction to and subsequent immersion in the theory of computer science, without whose influence I might not have found a great passion.

# Chapter 1

# Introduction

## 1.1 What Is a Cluster Graph, and What Is It Good for?

This section is an overture to those, like my parents, who begin to read this thesis with no prior knowledge of the topics to be discussed. The reader familiar with graph theory and cluster graphs in particular may wish to skip ahead to Section 1.2.

A graph is the purest and the simplest formalism used to answer the question, "Which among you are connected in this way?" It consists of two parts: a set of things and a set explaining which pairs among those things are related. When we observe a pairwise relation between objects in nature, we might formalize our observations as a graph and use existing knowledge in the field of graph theory to inform our inferences about those objects. Conversely, we might seek more precise and profound results in graph theory in the hope that they can be transmuted into insight about those worldly objects that we model with graphs already.

Multitudes of real-world problems yield readily to graph theory. In 1741, Leonhard Euler solved the question of whether or not the seven bridges of the city of Konigsberg could be crossed without repetition (they could not [15]). In doing so, he began a long tradition of using graphs to model physical locations and the ways to travel between them. This is as simple as selecting a set of places (cities, subway stations, islands, rooms), associating each with a vertex, and connecting each pair of vertices with an edge if it is possible to travel directly between the corresponding locations. Optionally, one can add weights to the edges to represent distance or difficulty of travel. The result is a conveniently simple, fruitfully abstract model of the world that begs to be reasoned about.

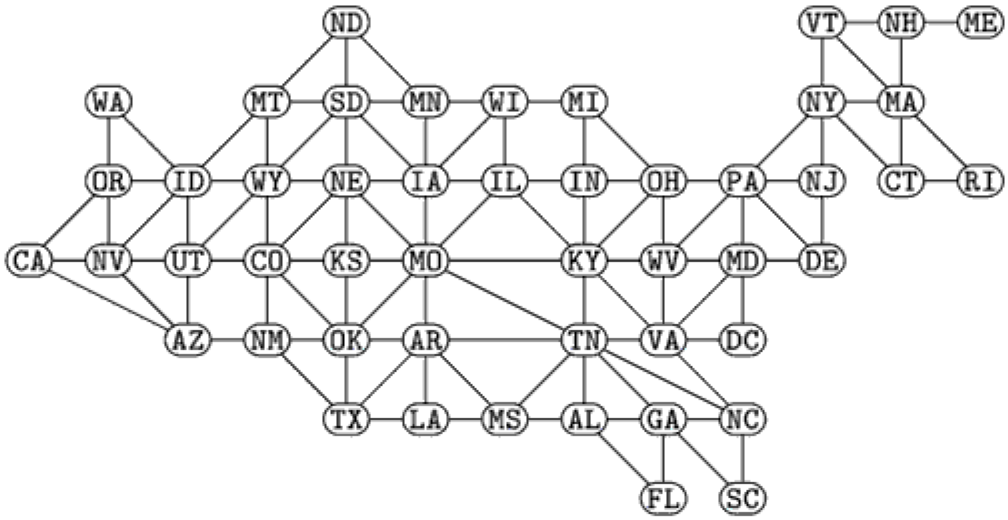



Figure 1.1: The "Contiguous USA Graph," in which each state vertex is connected by an edge to the state vertices with which it shares a land border. Reproduced from [21].

Some of the most famous problems in mathematics are accessible to this practice. For example, the Travelling Salesman Problem, or TSP, famously asks, "What is the shortest distance in which a salesman can travel to each of the following cities?" An instance of the TSP is most intuitively represented as graph: a set of cities and a set of distance-marked edges between them. An introduction to the Travelling Salesman Problem is provided by [9].

However, mutual accessibility is far from the only relation well-modeled by graphs. Consider the following (entity; relation) pairs:

- people; bosom friends
- people; mortal enemies
- websites; hyperlinked pairs
- wireless towers; tower pairs in broadcast range
- syntactic objects; constituents and root nodes
- individual atoms; bonded pairs
- airports; those with direct connecting flights
- neurons; those connected by axa in a brain

- neural nodes; those connected in an artificial neural network
- mathematicians; collaborators
- bioregions; those between which migration is possible

Complex reasoning about each of these domains is possible using the tools of graph theory. The list above is far from complete.

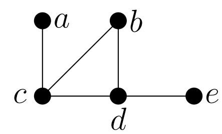



Figure 1.2: A planar graph with five vertices.

Graphs are traditionally represented on the plane by drawing each vertex as a point and drawing a curve between each pair of points if they share an edge, as illustrated in Figure 1.2. To prevent confusion (and sometimes to manifest a limitation inherent in the system being modeled) it is often preferable to draw the graph on the plane in such a way that no edges cross. We call the class of graphs for which this is possible the *planar graphs*, and they are fairly easily distinguished from the nonplanar graphs. However, the task of coming up with the most intelligible representation of a nonplanar graph is much more difficult than the analogous task for a planar graph. At present, much energy is focused on discovering ways to render not-quite-planar graphs most usefully on the plane.

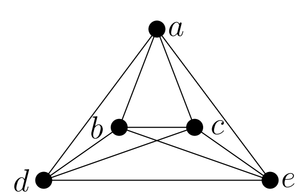



Figure 1.3: A nonplanar graph with five vertices.

One sensible approach for rendering nonplanar graphs is to simplify the structure of the graph by grouping "similar" vertices together. For this task, we turn to the concept of a partition. In contrast to a graph, which emphasizes pairwise relationships, a partition divides a set of things into groups that are in some way self-similar. Like graphs, examples of partitions in nature are innumerable. However, consider just a few (entity; grouping) pairs to spark the imagination:

- people; eye color
- counting numbers; parity
- skittles; flavor
- novelists; genre
- words; part of speech

The notion of a cluster graph combines a graph with a partition, and thus may represent a set of objects partitioned into groups and simultaneously linked by a pairwise relation. In a cluster graph, pairs of objects are connected by edges to indicate relation, and groups of objects are partitioned into clusters to indicate some common property. For example, Di Giacomo, Didimo, Liotta, and Palladino considered the graph of European scholars who had recently published in the journal *Graph Drawing*, connecting each pair with an edge if they had collaborated on research, and clustering groups based on their country of residence [11]. A completed cluster graph representation, reproduced in Figure 1.4, displays their collaboration both individually and internationally.

Furthermore, the partition of vertices into clusters need not be semantically distinct from the relation captured by the graph. In the physical sciences, clustering is often used to isolate meaningful structure implicit in a graph generated from experimental data. In this case, clustering is performed automatically by an algorithm trained to detect a particular type of structural commonality. For an accessible overview of the graph clustering problem, the interested reader is referred to [26].

This thesis unifies established methods for representing cluster graphs on the plane by introducing a new representation schema, the $(k, p)$-planar drawing, that is broad in scope and precise in specification. In a $(k, p)$-planar drawing, each cluster is represented by a single contiguous region, and each vertex in a cluster is associated with one or more access points, called *ports*, located on the perimeter of the region.

One final note for the lay reader, if she has made it this far. After the introduction, this thesis contains four main chapters, each of which is composed mostly of technical results. However, each chapter and each main section is preceded by a summary paragraph that can serve as an anchor for the reader cast adrift or a lighter substitute for the meat of each section.

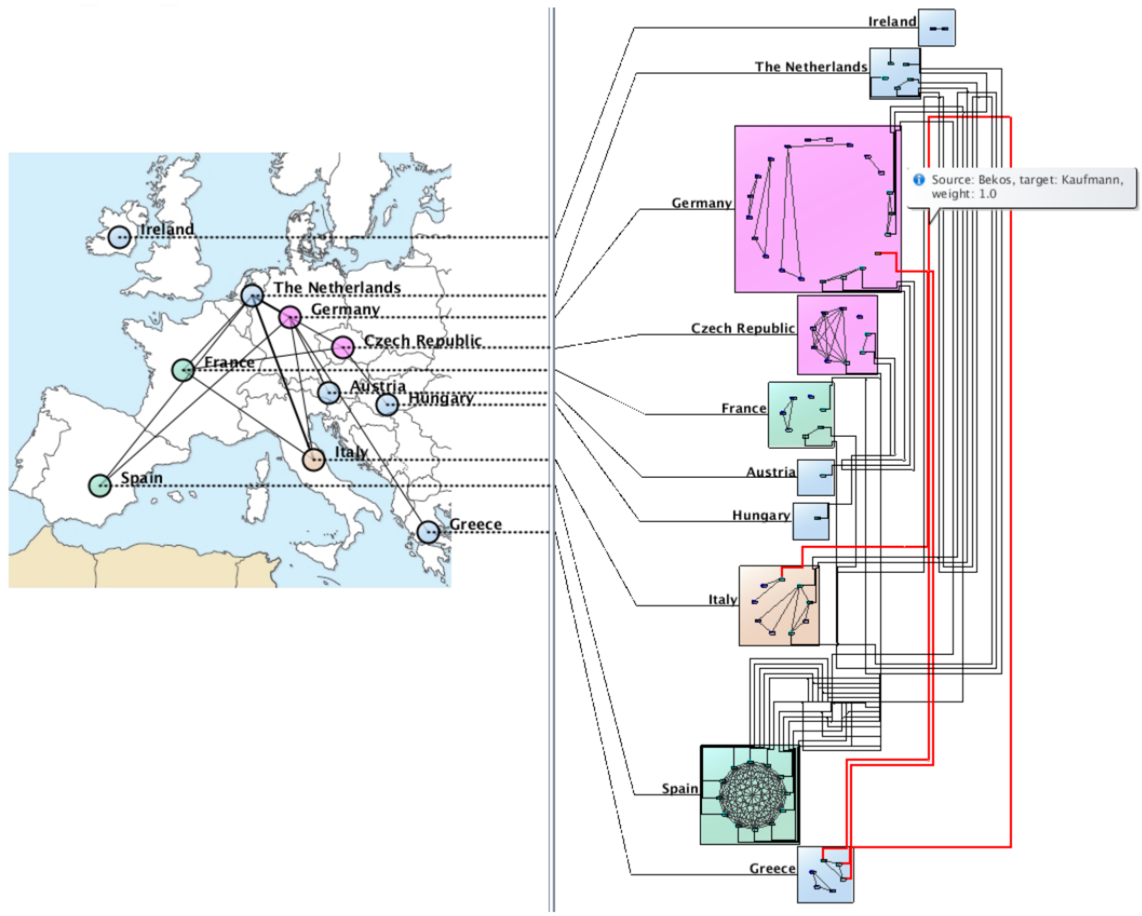



Figure 1.4: European coauthorship. Reproduced from [11].

## 1.2 Preliminary Definitions

A *graph* $G$ consists of a set $V$ of vertices and a set $E$ of edges. A *cluster graph* consists of a graph $G$ and a partition of the vertex set into *clusters* $V_1, V_2, ...V_C$. We call an edge *intracluster* if it connects two vertices in the same cluster and *intercluster* otherwise.

A graph is *k-clustered* if no cluster contains more than $k$ vertices.

A *port drawing* of a cluster graph G is a drawing $\Gamma$ of G with the following properties:

- Each cluster $V_i$ is represented by a distinct convex region $R_i$ called a *cluster region*. Cluster regions may not intersect or be drawn inside each other.

- Each vertex is represented by one or more distinct points, called *ports*, on the perimeter of its cluster region.

- Each intercluster edge $(u, v) \in E$ is represented by a simple curve connecting a port of $u$ to a port of $v$. Edges may not intersect cluster regions.

We say that a port drawing is *planar* if no intercluster edges cross.

This thesis considers the classes of $(k, p)$-*planar graphs*, the $k$-clustered graphs that admit planar port drawings with no more than $p$ ports per vertex. For simplicity, we will also use the term $(k, p)$-*planar* to refer to those unclustered graphs that can be $k$-clustered so that the resulting cluster graphs are $(k, p)$-planar.

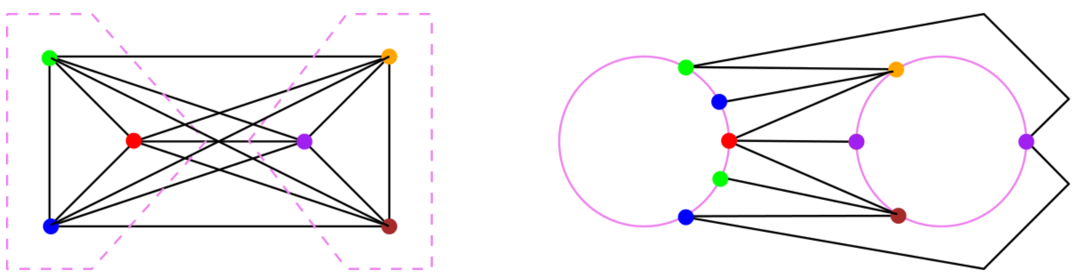


Figure 1.5: Node-link and $(3, 2)$-planar representations of the same graph.

Figure 1.5 compares node-link and $(3, 2)$-planar drawings of the same nonplanar graph $G$. The two clusters are outlined by pink dashes in the node-link drawing at left and represented by pink cluster regions in the $(3, 2)$-planar drawing at right. In the $(3, 2)$-planar drawing, planarity is achieved by dividing the green, blue, and purple vertices into two ports each.

## 1.3 Advantages and Applications of $(k, p)$-Planar Graphs

The first part of this section considers the advantages of the $(k, p)$-planar drawing as a planar representation schema for cluster graphs. The second part elaborates several specific applications that motivate the use of $(k, p)$-planar drawings.

### 1.3.1 General Advantages of $(k, p)$-Planar Drawings

This thesis is primarily motivated by the need to better represent the features of large nonplanar graphs, especially those that exhibit different properties at the global and local scales. $(k,p)$-planar drawings are particularly well suited to represent graphs that are globally sparse but have subgraphs with interesting properties, such as density, Hamiltonicity, or common membership in an externally defined category. In a $(k,p)$-planar drawing, global structure is preserved by intercluster edges and emphasized by the requirement that no intercluster edges may cross. The fact that many nonplanar graphs are $(k,p)$-planar for small values of $k$ is facilitated by the use of ports, which allow clustered vertices to be effectively split while remaining tightly associated with the other vertices in the same cluster.

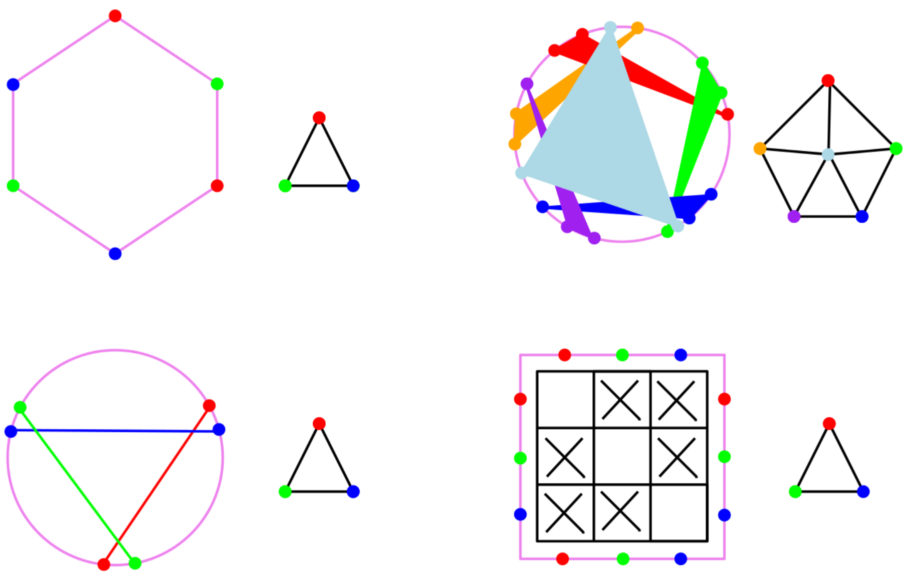


Figure 1.6: Examples of intracluster representations and corresponding node-link drawings. Counterclockwise from top left: no internal representation, inscribed circle graph, inscribed adjacency matrix, and inscribed polygon-circle graph.

The necessary consequence of the use of cluster regions is the omission of intracluster edges. However, because $(k,p)$-planar drawings leave the area inside the cluster region empty, various interior representations can be used to highlight intracluster structure. As illustrated in Figure 1.6, high-density clusters can be realized by a variety of intracluster representations that display them with greater readability than traditional node-link drawings. In addition, when the vertices of a graph are partitioned on the basis of an external representation, such as social group or geographic location, cluster regions can be annotated to indicate the common feature of their constituent vertices. Finally, cluster regions corresponding to subgraphs with particular topological features can be inscribed with representations that highlight their structure. The reader is referred to Chapter 5 for a survey of intracluster representations.

A final advantage of the $(k, p)$-planar graph is its generality. Section 1.4 describes the way in which $(k, p)$-planar graphs generalize and incorporate the advantages of many existing planar representations for cluster graphs.

To illustrate why better planar representations for cluster graphs are necessary, the following three subsections consider graph applications which provide challenges for existing representations.

### 1.3.2 Small World Networks

*Small-world networks*, introduced in 1998 by Watts and Strogatz [28], are graphs that are globally sparse but locally dense. They occur naturally in the analysis of social networks and graphs of interlinked webpages, among other domains. Consider for example a graph $G$ of friend relationships between users of a large social network. The chance that any two randomly sampled users are friend-related is small, and thus $G$ is globally sparse. However, certain subgraphs, such as those corresponding to families, groups of close friends, and professional organizations, may be almost completely connected. The interested reader is referred to [24] for an explanation of the phenomena underlying small-world networks and additional information on network structure.

The small-world network presents a problem for traditional representations. If the vertices of a community subgraph are represented near each other, the high edge density of the subgraph will make the specific relations within the community less clear. Alternatively, if the vertices of a community subgraph are separated, their tight-knit structure will be obscured by distance, while their numerous edges may confuse the global structure of the graph.

Clustering a small-world network by community creates a cluster graph that can be well-represented by a $(k, p)$-planar drawing. Within a cluster region, community relations can either be omitted or represented using an alternative representation more appropriate for a high-density subgraph. Moreover, the number of necessary intercluster edge crossings can be mitigated by distributing intercluster edges between multiple ports.

### 1.3.3 Graphs with Mined Substructures

*Substructure mining* refers to the problem of locating topologically significant subgraphs within large graphs, usually graphs of experimental data. Mined subgraphs include paths and cycles [1, 2], spanning trees [16], and subgraphs with high frequency [8, 29].

Once located, these substructures must be effectively represented. Although mined substructures may not be particularly dense, an effective representation should display them in a way that highlights their features and distinguishes them from the larger graph. For certain types of subgraphs, such as cycles and trees, representations that highlight their topology may be preferable. Alternatively, when an algorithm identifies a structural feature that occurs frequently within subgraphs, an appropriate representation should focus on this feature. Both of these goals can be accomplished in a $(k, p)$-planar graph by clustering mined subgraphs and choosing appropriate intracluster representations.

### 1.3.4 External Partition Graphs

Instead of arising from the structure of the graph itself, as in the cases of small world networks and graphs with mined substructures, the partition of a cluster graph may be only tangentially related or completely independent from the edge-relation. For example, recall the cluster graph presented in [11] and reproduced as Figure 1.4, in which author vertices are related by coauthorship and partitioned by nationality.

The representation of an external partition graph shares some desiderata with those of topological feature graphs and small-world networks. Certain cluster subgraphs might be very dense, in which case alternative representations such as intersection graphs will be helpful. Other clusters, perhaps in the same graph, might be sparse or display structural similarities that ought to be highlighted as in feature graphs. In the case of an external partition, it is particularly important that cluster representations are contiguous or otherwise unified. $(k, p)$-planar graphs satisfy each of these requirements.

## 1.4 Previous Scholarship

This section summarizes proposed representations for cluster graphs, evaluates their use cases, and explains their reduction to $(k, p)$-planar graphs if applicable. The various representations demonstrate the trade-offs between emphasizing global and local features, minimizing edge crossings, and reducing information loss.

### 1.4.1 NodeTrix Representations

The NodeTrix framework, presented by Henry, Fekete and McGuffin in [18], is a representation for cluster graphs intended specifically to highlight the local intricacies and global structure of small-world graphs. NodeTrix represents clusters by drawing their corresponding adjacency matrices in the plane. In a NodeTrix representation, each intercluster edge $(u, v)$ is represented by a curve drawn between an adjacency matrix row or column corresponding to $u$ and an adjacency matrix row or column corresponding to $v$. An example NodeTrix representation, reproduced from [18], is illustrated in Figure 1.7.

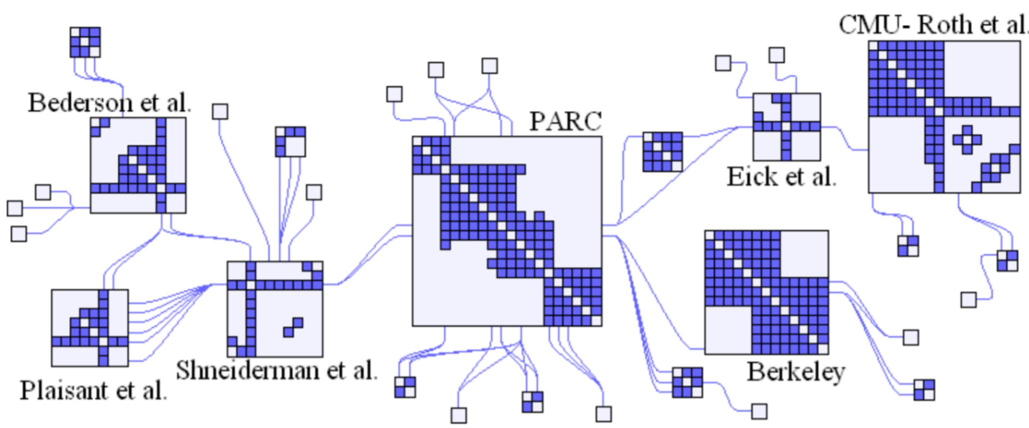



Figure 1.7: A NodeTrix representation of a coauthorship graph. Reproduced from [18].

Several authors have employed, analyzed and extended the NodeTrix framework [10, 17, 20]. Di Giacomo et al. [13] define an $n$-NodeTrix-planar graph as a graph that admits a NodeTrix representation with matrices of dimension at most $n$ and no crossing edges. $n$-NodeTrix-planar graphs are readily interpreted as $(k, p)$-planar graphs. From the observation that each row and column end can be treated as a port, it follows that a 2-NodeTrix-planar representation is equivalent to a $(2, 3)$-planar drawing, and that an $n$-NodeTrix-planar drawing is equivalent to a $(n, 4)$-planar drawing for any fixed integer $n$ greater than 2.

### 1.4.2 $(X,\ Y)$-Clustering Representations

In [4], Batagelj et al. explore the possibility of representing cluster graphs using a method called *(X, Y)-Clustering*. An $(X,\ Y)$-Clustering of a graph is a clustering such that the graph obtained by contracting each cluster has property $X$ and each cluster subgraph has property $Y$. Figure 1.8, reproduced from [4], illustrates a (planar, 4-clique)-clustered graph.

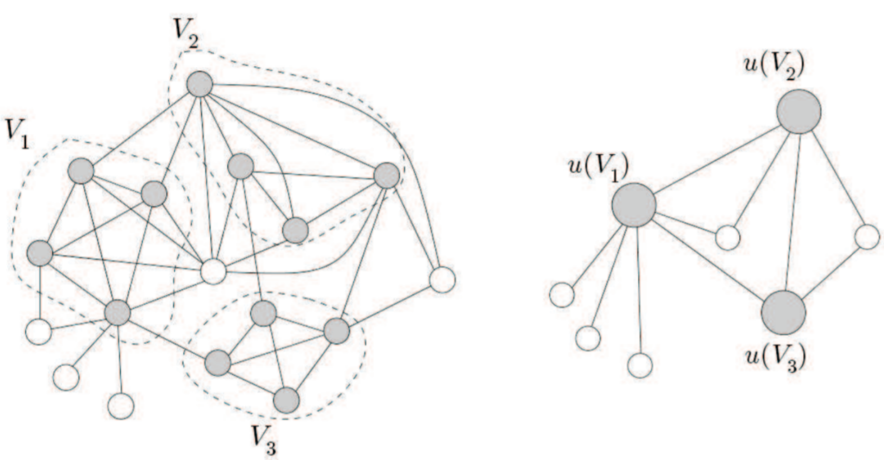



Figure 1.8: Node-link and contracted representations of a (planar, 4-clique)-clustered graph. Reproduced from [4].

The authors present $(X,\ Y)$-clustering as a representation schema in conjunction with an interactive system, in which the user is first presented with a node-link representation of the graph obtained by contracting each cluster and then may click each contracted cluster vertex to view the intracluster structure. As such, $(X,\ Y)$-clustering does not imply a particular planar representation for cluster graphs for any fixed $X$ and $Y$. However, as demonstrated by Figure 1.8, the contracted node-link representation of a (planar, $Y$)-clustered graph is equivalent to a $(k, p)$-planar graph for some $p$ if each contracted vertex is converted to a cluster region.

### 1.4.3 Intersection-Link Representations

In [3], Angelini et al. introduce the *intersection-link* representation, a planar representation for cluster graphs that represents intercluster edges as traditional links and cluster subgraphs as the intersection graphs of rectangles. Figure 1.9, reproduced from [3], illustrates an intersection-link representation of a graph partitioned into cliques.

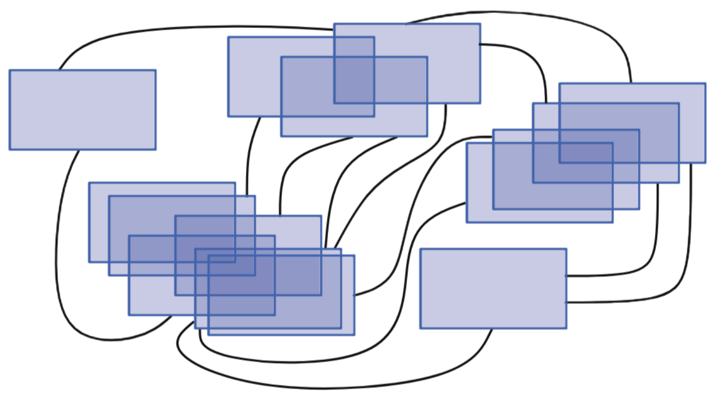


Figure 1.9: An intersection-link representation of a graph partitioned into cliques. Reproduced from [3].

Like the $(k, p)$-planar drawing, the intersection-link representation leverages the insight that good representations for small world networks can be achieved by representing intercluster edges as traditional links and using alternate representations for dense subgraphs. However, intersection-link representations are more restrictive than $(k, p)$-planar graphs. The original formulation of the intersection-link representation requires that each cluster be a clique and that each vertex be represented by an identical rectangle. Even if the clique requirement is relaxed, the identical rectangle requirement ensures that each rectangle can have at most 4 separate exposed perimeter sections, with the result that every intersection-link representation is equivalent to a $(k, 4)$-planar drawing.

### 1.4.4 Vertex Splitting Representations

Vertex splitting allows a very different sort of planar representation than those previously discussed. Following the convention of Eppstein et al. [14], we define the $k$-split operation as the replacement of a single vertex $v$ in a graph with $k$ new vertices such that each neighbor of $v$ is adjacent to exactly one newly created vertex. We say that a graph is $k$-splittable if it can be transformed into a planar graph by $k$-splitting some subset of the vertices.

A planar drawing of the graph resulting from a series of $k$-splits may be regarded as a planar port representation in which the ports of each vertex are not required to be located on the perimeter of a cluster region. The vertex splitting approach has the advantage of flexibility, but it is a poor choice for a cluster graph representation. Although vertices may be colored and labeled, a planar $k$-split drawing specifies no spatial connection between vertices resulting from a split or vertices in the same cluster, thus obscuring the structure of the original graph.

## 1.5 Summary of Results

The remainder of this thesis is organized as follows.

Chapter 2 determines relates the classes of $(k, p)$-planar graphs to established graph classes. We prove that the $(k, 1)$-planar graphs are equivalent to the planar graphs in the $k \leq 3$ case, and that the $(4, 1)$-planar graphs are equivalent to the IC-planar graphs. We further prove that there exists a $(5, 1)$-planar graph that is not NIC-planar and a $(6, 1)$-planar graph that is not 1-planar. We define the class of TrNIC-planar graphs and prove that while for any fixed $k$ there exist AcNIC and TrNIC-planar graphs that are not $(k, 1)$-planar, the classes of AcNIC and TrNIC-planar graphs are subsets of the class of $(2, 2)$-planar graphs. Finally, we prove that that not all $(2, 2)$-planar graphs are NIC-planar and not all NIC-planar graphs are $(2, 2)$-planar.

Chapter 3 bounds the edge density of the $(k, p)$-planar graphs. In the first part of the chapter, we prove two edge bounds parameterized by a given number $C$ of clusters. In particular, we prove that the number of intercluster edges in a $(k, p)$-planar graph with at least three vertices is at most $(3C - 6)k^2$ and at most $(kp + 3)C - 6$. Which bound is smaller depends on the parameters $k$ and $p$. In addition, we prove that the first bound is tight when $p > 3k$ and that the second bound is tight when $p < k$ and $k > 1$. Corollary to these results, we establish tight bounds on the total number of edges in a $(k, p)$-planar graph with $C$ clusters.

The first two bounds presented in Chapter 3 are helpful for determining whether a $k$-clustered graph is too dense to be $(k, p)$-planar. However, to determine the maximum edge density of any $(k, p)$-planar graph on $|V|$ vertices, independent of the number of clusters, more work is required. In the second part of the chapter, for any $(k, p)$-planar graph $G = (V, E)$, we prove the bound

$$|E| \leq \sum_{i=2}^{k}(c_i \cdot (ip + 3 + \frac{i(i-1)}{2})) + 3c_1 - 6, \tag{1.1}$$

where $c_i$ is the number of clusters of size $i$. Finally, we maximize Equation 1.1 over all possible clusterings to generate a tight bound for the maximum number of edges in a $(k, p)$-planar graph with $|V|$ vertices.

Chapter 4 focuses on the hardness of the $(k, p)$-planarity decision problem: given a graph $G$ and fixed values for $k$ and $p$, is $G$ a $(k, p)$-planar graph? Because the $(k, 1)$-planar graphs are planar when $k \leq 3$, and planarity is testable in linear time, deciding $(k, 1)$-planarity is a linear time problem when $k \leq 3$. Because the $(4, 1)$-planar graphs are equivalent to the IC-planar graphs, and deciding if a graph is IC-planar is an NP-complete problem [7], deciding $(4, 1)$-planarity is NP-complete. In addition, we prove that the $(k, 1)$-planarity problem can be decided in linear time for all $k$ if a clustering is specified. Finally, we provide a proof that deciding $(2, 2)$-planarity is NP-complete. Table 1.1 summarizes the hardness results proved in Chapter 4.

| p \ k | 1 | 2 | 3 | 4 | ≥5 |
|---|---|---|---|---|---|
| 1 | in $\mathbb{P}$ | in $\mathbb{P}$ | in $\mathbb{P}$ | $\mathbb{NP}$-complete | ? |
| 2 | in $\mathbb{P}$ | $\mathbb{NP}$-complete | ? | ? | ? |
| $\geq 3$ | in $\mathbb{P}$ | ? | ? | ? | ? |

Table 1.1: Hardness of deciding $(k, p)$-planarity.

Chapter 5 considers extensions of $(k, p)$-planarity which specify various intracluster representations. We say that a graph $G$ is $(k, p)$-$X$-planar if it admits a $(k, p)$-planar drawing in which the interior of each cluster region represents its intracluster structure according to some representation $X$. For instance, a graph is $(k, 2)$-circle-planar if it admits a $(k, 2)$-planar drawing in which each cluster region is a circle and the interior of each cluster region is represented as a circle graph. We also consider more permissive intracluster representations.

Chapter 6 concludes the thesis by considering the broader implications of our results for the potential of the $(k, p)$-planar graph and discussing possible avenues for future work.

# Chapter 2

# Relating the $(k, p)$-Planar Graphs to Other Graph Classes

In this chapter, we examine the relationship between the $(k, p)$-planar graphs, the planar graphs, and several classes of nonplanar graphs, including the IC-planar, AcNIC-planar, NIC-planar, and 1-Planar graphs. We also introduce the class of TrNIC-planar graphs, a graph class that generalizes the AcNIC-planar graphs and specializes the NIC-planar graphs.

We begin the chapter by proving that in the $k = 1$, $k = 2$, and $k = 3$ cases, the $(k, 1)$-planar graphs are equivalent to the planar graphs. We then prove that the $(4, 1)$-planar graphs are equivalent to the IC-planar graphs. Beyond this point, the relationship between the $(k, 1)$-planar graphs and established nonplanar graph classes breaks down. We note that the class of $(5, 1)$-planar graphs contains the complete graph $K_6$, which is not NIC-planar, and that the class of $(6, 1)$-planar graphs contains the complete graph $K_7$, which is not 1-planar. However, we prove that there exists an AcNIC-planar graph that is not $(k, 1)$-planar for any fixed integer $k$.

If we allow each vertex multiple ports, the task of relating the $(k, p)$-planar graphs to established nonplanar graph classes becomes more complicated. We prove that the TrNIC-planar graphs are a proper subset of the $(2, 2)$-planar graphs. To conclude the chapter, we prove that the classes of NIC-planar graphs and $(2, 2)$-planar graphs are overlapping but distinct: there exist NIC-planar graphs that are not $(2, 2)$-planar, and $(2, 2)$-planar graphs that are not NIC-planar.

Before beginning, we note that the classes of $(k, p)$-planar graphs are naturally related to each other by the following proposition.

**Proposition 1.** *For any positive integers $k_0$, $k_1$, $p_0$, and $p_1$ with $k_0 \leq k_1$ and $p_0 \leq p_1$, the class of $(k_0,p_0)$-planar graphs is a subset of the class of $(k_1,p_1)$-planar graphs.*

*Proof.* Let $k_0$, $k_1$, $p_0$, and $p_1$ be positive integers such that $k_0 \leq k_1$ and $p_0 \leq p_1$, and let $G$ be a $(k_0,p_0)$-planar graph with $(k_0,p_0)$-planar drawing $\Gamma$. As $\Gamma$ is also a $(k_1,p_1)$-planar drawing by definition, $G$ is $(k_1,p_1)$-planar. Thus we have

$$(k_0,p_0)-Planar \subset (k_1,p_1)-Planar.$$

$\square$

## 2.1 Definitions for Nonplanar Graph Classes

In this section, we describe the established graph classes to which we will compare the $(k,p)$-planar graphs. These classes are related as follows.

$$Planar \subset IC-Planar \subset AcNIC-Planar \subset TrNIC-Planar \subset 1-Planar.$$

### 2.1.1 1-Planar Graphs

The 1-planar graphs, the largest class of nonplanar graphs we consider in this chapter, are defined as follows.

**Definition 1.** *A graph $G$ is 1-planar if it admits a drawing in which each edge crosses at most one other edge.*

The following subsections define nonplanar graph classes that specialize the 1-planar graphs.

### 2.1.2 IC-Planar Graphs

The smallest class specializing the 1-planar graphs that we consider is the class of Independent Crossing or IC-planar graphs [31]. They are formally defined as follows.

**Definition 2.** *An* IC-planar drawing *is a drawing in which each edge crosses at most one other edge and no vertex is adjacent to more than one crossing edge. A graph $G$ is* IC-planar *if it admits an IC-planar drawing.*

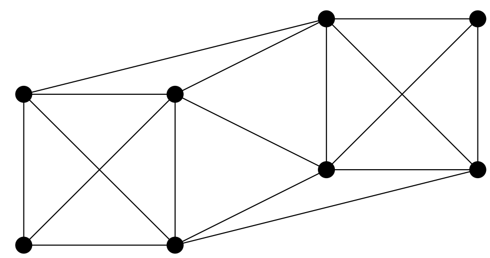


Figure 2.1: An IC-planar drawing with two disjoint crossing pairs.

We refer to a pair of crossing edges in a nonplanar drawing as a *crossing pair*, a helpful definition which allows us to specify the internal structure of several subsequent graph classes. Figure 2.1 illustrates an IC-planar drawing with two crossing pairs.

We define the *crossing-pairs graph* of a drawing Γ as follows.

**Definition 3.** *The crossing-pairs graph, or* cp-graph*, of a drawing* Γ *is the graph that has a vertex for each crossing pair in* Γ *and an edge between each pair of vertices if their corresponding crossing pairs share a vertex.*

### 2.1.3 AcNIC-Planar Graphs

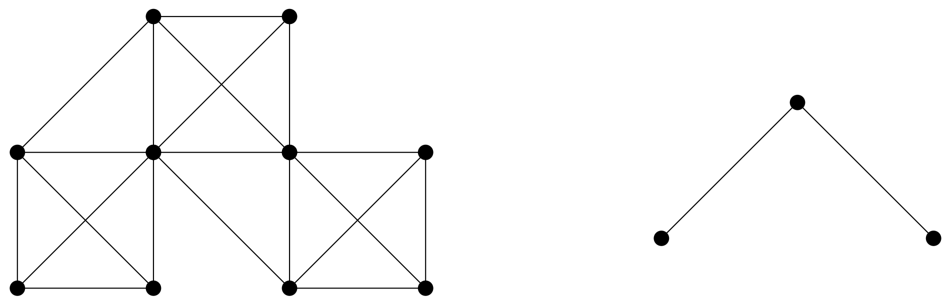
The AcNIC-planar graphs, introduced by Di Giacomo, Liotta, and Tappini in [12], generalize the IC-planar graphs and specialize the 1-planar graphs. We define the AcNIC-planar graphs as follows.

**Definition 4.** *An* AcNIC-planar drawing *is a 1-planar drawing with an acyclic cp-graph in which no two crossing pairs share more than one vertex. A graph G is AcNIC-planar if it admits an IC-planar drawing.*

Figure 2.2 displays an AcNIC-planar drawing and its corresponding acyclic cp-graph.

### 2.1.4 TrNIC-Planar Graphs

The class of AcNIC-planar graphs can be further generalized to the class of *TrNIC-planar* graphs, the class of nonplanar graphs whose crossing pairs form a treelike



(a) An AcNIC-planar drawing $\Gamma$ with three adjacent crossing pairs.

(b) The cp-graph associated with $\Gamma$.

Figure 2.2: An AcNIC-planar drawing and associated cp-graph.

structure.

The treelike structure underlying a TrNIC-planar drawing is captured by its *cp-cut-graph*, which is constructed from a planar drawing $\Gamma$ by creating a vertex $v_{cp}$ for each crossing pair in $\Gamma$ and a hub vertex $v_h$ for each vertex in $\Gamma$ adjacent to two or more crossing pairs. The cp-cut-graph contains the edge $(v_h, v_{cp})$ if $v_h$ and $v_{cp}$ correspond to an adjacent vertex and crossing pair in $\Gamma$. We define the TrNIC-planar graphs in terms of the cp-cut-graph as follows.

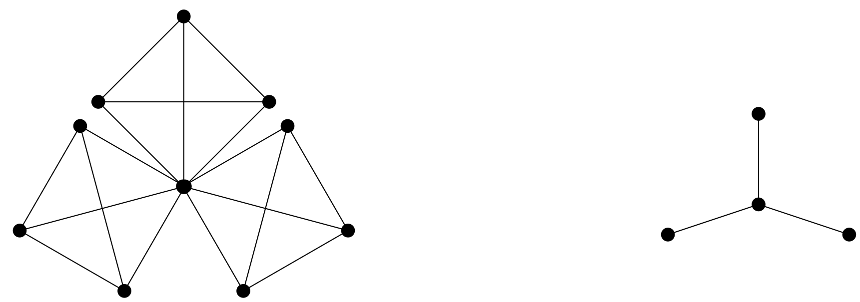
**Definition 5.** *A graph $G$ is* TrNIC-planar *if it admits a 1-planar drawing $\Gamma$ with acyclic cp-cut-graph.*

Figure 2.3 illustrates a TrNIC-planar drawing and corresponding cp-cut-graph. Note that the graph depicted in Figure 2.3 is not AcNIC-planar, as its cp-graph is a 3-cycle.

### 2.1.5 NIC-Planar Graphs

The class of Nearly IC-planar graphs, or NIC-planar graphs, generalizes the TrNIC-planar graphs and specializes the 1-planar graphs. It has the following definition.

**Definition 6.** *A graph $G$ is NIC-planar if it admits a* NIC-planar drawing, *a drawing $\Gamma$ in which no two crossing pairs share more than one vertex.*



(a) A TcNIC-planar drawing $\Gamma$ with three adjacent crossing pairs.

(b) The cp-cut-graph associated with $\Gamma'$.

Figure 2.3: A TrNIC-planar drawing and associated cp-cut-graph.

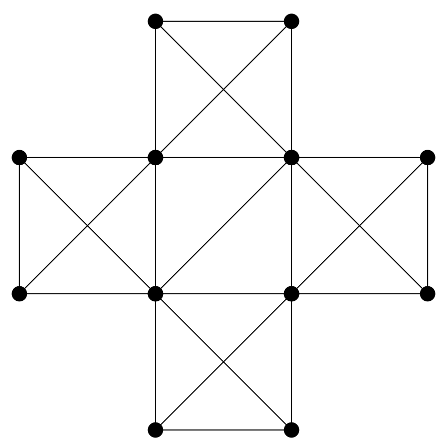


Figure 2.4: A NIC-planar drawing.

Figure 2.4 illustrates a NIC-planar drawing. Note that the cp-graph and the cp-cut-graph of the graph depicted in Figure 2.4 are both cycles, so the graph is neither AcNIC-planar nor TrNIC-planar.

With these definitions in hand, we consider the $(k, 1)$-Planar graphs.

## 2.2 Relating $(k, 1)$-Planar Graphs to Established Graph Classes

In this section, we prove that the classes of $(1, 1)$-planar, $(2, 1)$-planar, and $(3, 1)$-planar graphs are equivalent to the class of planar graphs, and that the class of $(4, 1)$-planar graphs is equivalent to the class of IC-planar graphs. In the $k > 4$ case, the neat correspondence between the $(k, 1)$-planar graphs and established classes of nonplanar graphs degrades. We show that there exists a $(5, 1)$-planar graph that is not NIC-planar, that there exists $(6, 1)$-planar graph that is not 1-planar, and that for any fixed positive integer $k$, there exists an AcNIC-planar graph that is not $(k, 1)$-planar.

### 2.2.1 Planar $(k, 1)$-Planar Graphs

**Theorem 1.** *For $k \leq 3$, a graph $G$ is $(k, 1)$-planar if and only if it is planar.*

*Proof.* The necessity condition of this proof is trivial, as every planar graph is $(1, 1)$-planar and thus $(k, 1)$-planar for any positive integer $k$ by Proposition 1.

To establish the sufficiency condition, let $\Gamma$ be a $(3, 1)$-planar drawing of a graph $G$. Replace the ports of each cluster region in $\Gamma$ with their corresponding vertices, draw intracluster edges as necessary, and retain the intercluster edges from $\Gamma$ to generate a planar drawing of $G$.

Thus any graph with a $(3, 1)$-planar drawing is planar. Because the $(2, 1)$-planar and $(1, 1)$-planar graphs are $(3, 1)$-planar by Proposition 1, this suffices to establish Theorem 1. □

Corresponding $(3, 1)$-planar and planar drawings of a graph $G$ are depicted in Figure 2.5.

### 2.2.2 IC-Planar $(k, 1)$-Planar Graphs

By replacing the four vertices of a crossing pair with a cluster region, we can effectively remove the crossing without otherwise changing the drawing in which the crossing pair is embedded. This insight results in a procedure for representing IC-planar graphs with $(4, 1)$-planar drawings, and allows us to prove the following theorem.

**Theorem 2.** *A graph $G$ is $(4, 1)$-planar if and only if it is IC-planar.*

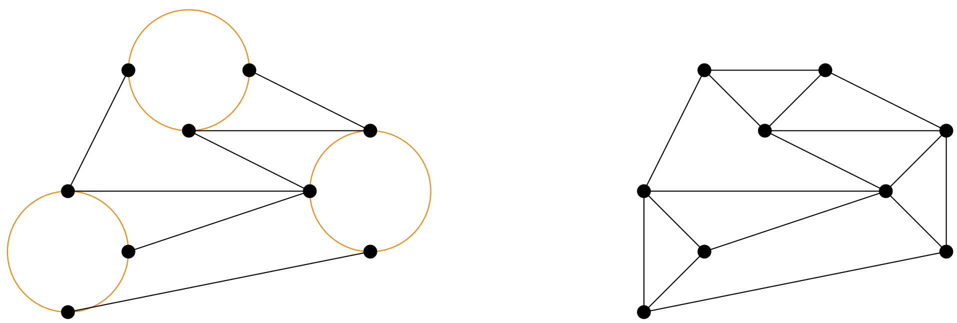
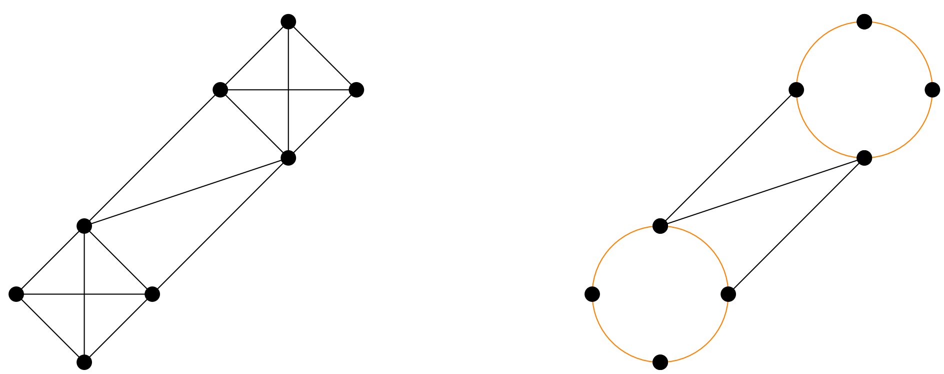


Figure 2.5: Corresponding $(3,1)$-planar and planar drawings.

*Proof.* First, we prove that the IC-planar graphs are $(4,1)$-planar. Given any IC-planar graph $G$, let $\Gamma$ be an IC-planar drawing of $G$. Replace each crossing pair with a 4-cluster region to create a $(4,1)$-planar drawing $\Gamma'$ of G. The application of this process transforms the IC-planar drawing illustrated in Figure 2.6a into the $(4,1)$-planar drawing illustrated in Figure 2.6b.

Because $\Gamma$ is an IC-planar drawing, no vertex in $\Gamma$ is adjacent to more than one crossing pair, and our clustering places no vertex in more than one cluster. $\Gamma'$ is thus a $(4,1)$-planar drawing.



(a) An IC-planar drawing of a graph $G$. (b) A $(4,1)$-planar drawing of a graph $G$.

Figure 2.6: Corresponding $(4,1)$-planar and IC-planar drawings.

Second, we prove that the $(4,1)$-planar graphs are IC-planar. Given any $(4,1)$-planar graph $G$, consider a $(4,1)$-planar drawing $\Gamma$ of $G$. Replace the ports of each cluster region with their corresponding vertices and draw intracluster edges between the new vertices as necessary to create a drawing $\Gamma'$. This process transforms the $(4, 1)$-planar drawing illustrated in Figure 2.6b back into the IC-planar drawing illustrated in Figure 2.6a. Because at most four vertices correspond to any cluster in $\Gamma$, drawing the intracluster edges of $\Gamma'$ on the plane creates at most one necessary crossing per cluster. Because no vertex is located in more than one cluster region in $\Gamma$, no vertex in $\Gamma'$ is adjacent to more than one crossing pair. $\Gamma'$ is thus an IC-planar drawing. □

### 2.2.3 Relating $(k, 1)$-Planar Graphs when $k > 4$

When $k > 4$, the $(k, 1)$-planar graphs exceed the nonplanar graph classes considered in this chapter. However, for arbitrarily large values of $k$, there remain graphs that are not $(k, 1)$-planar. In this section, we prove that there exists a $(5, 1)$-planar graph that is not NIC-planar, that there exists a $(6, 1)$-planar graph that is not 1-planar, and that for any fixed positive integer $k$, there exists an AcNIC-planar graph that is not $(k, 1)$-planar.

**Proposition 2.** *There exists a $(5, 1)$-planar graph that is not NIC-planar.*

*Proof.* Zhang proves in [30] that $K_6$ is not NIC-planar. However, $K_6$ is $(5, 1)$-planar as illustrated by Figure 2.7. Thus the $(5, 1)$-planar graphs are not a subset of the NIC-planar graphs. □

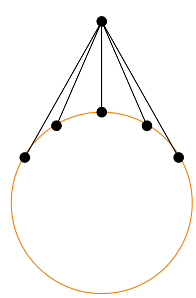


Figure 2.7: $K_6$ is $(5, 1)$-planar.

As a general rule, we observe that the complete graph $K_n$ is always $(n - 1, 1)$ planar. To create a $(n - 1, 1)$-planar drawing of $K_n$, we cluster $n - 1$ vertices and connect their ports to the single remaining vertex. To demonstrate the continued divergence of the $(k, 1)$-planar graphs from established nonplanar graph classes, we consider a second application of this principle.

**Proposition 3.** *There exists a* $(6, 1)$*-planar graph that is not 1-planar.*

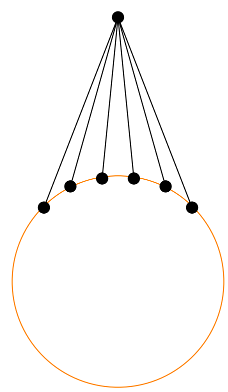


Figure 2.8: $K_7$ is $(6, 1)$-planar.

*Proof.* Korzhik proves in [23] that the graph $K_7 - K_3$, the complete graph on seven vertices with the edges of a 3-cycle removed, is not 1-planar, from which it follows that $K_7$ is not 1-planar. However, $K_7$ is $(6, 1)$-planar, as illustrated by Figure 2.8. □

However, no matter how large $k$ gets, some graphs are not $(k, 1)$-planar. Although all IC-planar graphs are $(4, 1)$-planar by Theorem 2, for any fixed $k$, some AcNIC-planar graphs (and thus some TrNIC-planar, NIC-planar, and 1-planar graphs) are not $(k, 1)$-planar. The following lemma is necessary for the proof of this result.

**Lemma 1.** *Let $G$ be a graph and let $K$ be a $K_5$ subgraph of $G$. Any* $(k, 1)$*-planar drawing $\Gamma$ of $G$ includes at least four vertices of $K$ in the same cluster.*

*Proof.* Assume for contradiction that there exists a $(k, 1)$-planar drawing $\Gamma$ of $G$ in which each cluster includes at most 3 vertices of $K$. Thus we can remove every port except those corresponding to vertices in $K$ from $\Gamma$ to generate a $(3, 1)$-planar drawing of $K_5$.

Our assumption implies that $K_5$ is $(3, 1)$-planar, which by Theorem 1 entails that $K_5$ is a planar graph. This creates a contradiction as $K_5$ is nonplanar. Thus any

$(k,1)$-planar drawing that includes a $K_5$ subgraph must cluster at least four vertices of the subgraph together. □

Lemma 1 enables us to establish the following theorem.

**Theorem 3.** *For any fixed positive integer $k$, there exists an AcNIC-planar graph that is not $(k,1)$-planar.*

*Proof.* First, we describe a family of graphs. Let the graph $G_n$ consist of a vertex $v$ fully connected to $n$ adjacent $K_4$ subgraphs as illustrated in Figure 2.9, which depicts an AcNIC-planar drawing of $G_5$. Note that we can extend this drawing arbitrarily by appending additional $K_4$ subgraphs to the end of the chain to create an AcNIC-planar drawing of $G_n$ for any positive integer $n$.

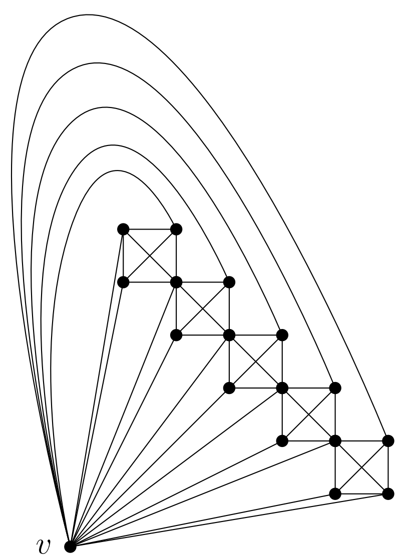



Figure 2.9: An AcNIC-planar drawing of the graph $G_n$ in the $k = 5$ case.

Furthermore, the subgraph induced by the vertices of each $K_4$ subgraph and $v$ is a $K_5$ subgraph. $G_n$ can thus be thought of as a chain of $K_5$ subgraphs, each of which shares two vertices with each of its neighbors. By Lemma 1, any $(k,1)$-planar clustering of $G_n$ must cluster four vertices from each of the $n$ adjacent $K_5$ subgraphs. For simplicity, we say that a cluster $C$ *covers* a $K_5$ subgraph if it contains at least four vertices from the subgraph. Thus it follows from Lemma 1 that any $(k,1)$-planar clustering of $G_n$ must cover each $K_5$ subgraph with some cluster.

However, because each $K_5$ subgraph shares two vertices with each of its neighbors, it is impossible to cover two adjacent $K_5$ subgraphs in $G_n$ with different clusters. Thus in any $(k,1)$-planar drawing of $G_n$ every $K_5$ subgraph in $G_n$ must be covered by the same cluster $C$. For a given $G_n$, this condition can be achieved by a cluster of size $2n+2$ that includes $v$, each of the $n-1$ vertices shared by two $K_5$ subgraphs, and an additional $n+2$ vertices selected from $G$ as necessary. Such a clustering takes full advantage of the vertices shared between multiple $K_5$ subgraphs, and thus no more efficient covering is possible.

Thus any $(k,1)$-planar drawing of $G_{\lceil k/2 \rceil}$ requires some cluster which contains more than $k$ vertices. For any fixed $k$, $G_{\lceil k/2 \rceil}$ is therefore an AcNIC-planar graph that is not $(k,1)$-planar. □

## 2.3 Relating $(k,2)$-Planar Graphs to Established Graph Classes

In the $p=2$ case, we can prove no precise equivalencies between the classes of $(k,p)$-planar graphs and the AcNIC-planar, TrNIC-planar, NIC-planar, and 1-planar graph classes. In this section, we show that the TrNIC-planar graphs, and thus the IC-planar and AcNIC-planar graphs, are a subset of the $(2,2)$-planar graphs. We further show that there exist NIC-planar graphs, and thus 1-planar graphs, that are not $(2,2)$-planar, but there also exist $(2,2)$-planar graphs that are not 1-planar, and thus not NIC-planar.

### 2.3.1 Relating $(2,2)$-Planar Graphs

In the following proof, we employ the *muffin gadget*, the $(2,2)$-planar drawing of a crossing pair depicted in Figure 2.10.

We can convert many nonplanar drawings into $(2,2)$-planar drawings by replacing crossing pairs with muffin gadgets. For example, any IC-planar drawing $\Gamma$ can be converted into a $(2,2)$-planar drawing by replacing each crossing pair with a muffin gadget as illustrated in Figure 2.11. Because unnecessary edges can be omitted, this procedure works whether or not the subgraph induced by the vertices of a crossing pair is a complete $K_4$ or is missing some edges.

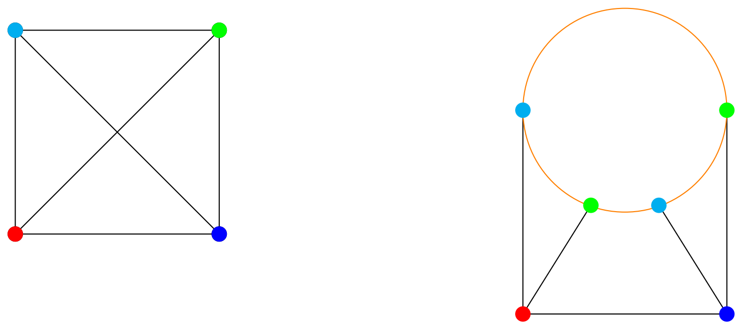


Figure 2.10: A crossing pair and corresponding $(2, 2)$-planar muffin gadget.

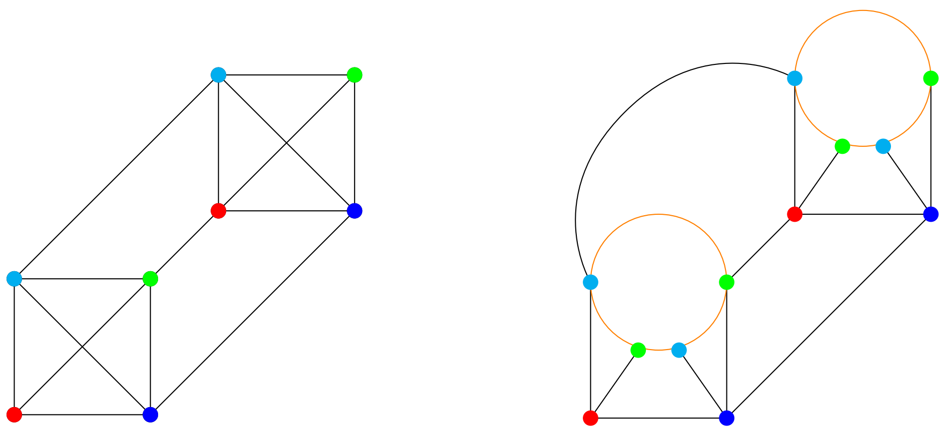


Figure 2.11: An IC-planar drawing and corresponding $(4, 1)$-planar drawing.

We employ a version of this method to prove that the TrNIC-planar graphs are a subset of the $(2, 2)$-planar graphs.

**Theorem 4.** *Every TrNIC-planar graph is* $(2, 2)$*-planar.*

*Proof.* Given a TrNIC-planar graph $G$, we demonstrate the construction of a $(2, 2)$-planar drawing of $G$, which suffices to prove Theorem 4.

Let $G$ be a TrNIC-planar graph, and let $\Gamma$ be a TrNIC-planar drawing of $G$ corresponding to the acyclic cp-cut-graph $G_{cp-cut}$. Replace each crossing pair in $\Gamma$ that corresponds to a leaf of $G_{cp-cut}$ with a muffin gadget, identifying the vertices of the muffin gadget with vertices of the crossing pair so that any vertex adjacent to multiple crossing pairs in $\Gamma$ remains unclustered. This is possible because each crossing pair in $\Gamma$ that corresponds to a leaf of $G_{cp-cut}$ shares at most one of its vertices with other crossing pairs in $\Gamma$.

Modify $G_{cp-cut}$ by deleting each leaf vertex corresponding to a replaced crossing pair. Then, delete any leaves in $G_{cp-cut}$ corresponding to hub vertices to produce the cp-cut-graph of the non-replaced crossing pairs remaining in $\Gamma$. By construction, the vertices of each non-replaced crossing pair in $\Gamma$ remain unclustered.

Finally, repeat the process of crossing pair replacement and vertex deletion. Because $G_{cp-cut}$ is a tree, each iteration of the process creates new leaves of $G_{cp-cut}$ until every crossing pair in $\Gamma$ has been replaced with a muffin gadget. The result is a drawing of $G$ that is $(2,2)$-planar. □

Because the TrNIC-planar graphs generalize the AcNIC-planar graphs, it follows from Theorem 4 that the AcNIC-planar graphs are also a subset of the $(2,2)$-planar graphs. However, some NIC-planar and 1-planar graphs are not $(2,2)$-planar. The following two results establish the existence of $(2,2)$-planar graphs that are neither NIC-planar nor 1-planar and NIC-planar and 1-planar graphs that are not $(2,2)$-planar.

**Theorem 5.** *There exists a $(2,2)$-planar graph that is not 1-planar.*

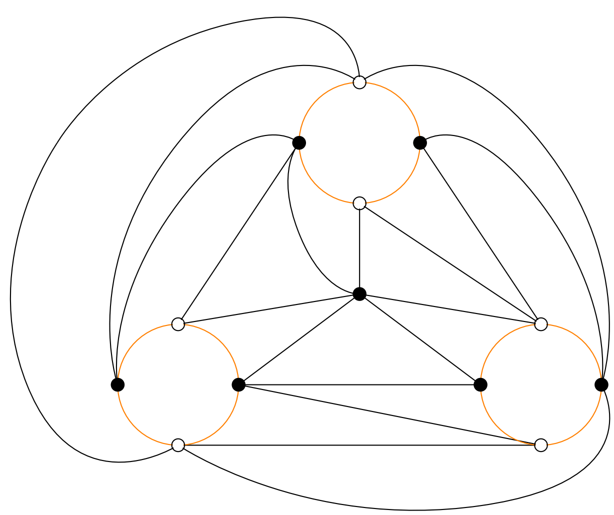


Figure 2.12: A $(2,2)$-planar drawing of $K_7$.

*Proof.* As noted in the proof of Proposition 3, the complete graph $K_7$ is not 1-planar. However, $K_7$ is $(2,2)$-planar as illustrated by Figure 2.12. □

As there exist $(2, 2)$-planar graphs that are not NIC-planar, and the class of NIC-planar graphs generalizes the TrNIC-planar graphs, the TrNIC-planar graphs are a proper subset of the $(2, 2)$-planar graphs.

**Theorem 6.** *There exists a NIC-planar graph that is not* $(2, 2)$*-planar.*

Theorem 6 follows from the result that there exist NIC-planar graphs that are not $(2, p)$-planar for any $p$, first proved by Tappini in [12].

## 2.4 Conclusion

In this chapter, we showed that for certain small values of $k$ and $p$, the classes of $(k, p)$-planar graphs are familiar. However, as $k$ and $p$ increase, the family of $(k, p)$-planar graphs burgeons. In particular, we showed that the classes of $(5, 1)$-planar graphs, $(6, 1)$-planar graphs and $(2, 2)$-planar graphs extend beyond the boundaries of the classes of NIC-planar and 1-planar graphs. The breadth of the class of $(k, p)$-planar graphs indicates the range of nonplanar graphs that can be represented by $(k, p)$-planar drawings.

# Chapter 3

# The Density of $(k, p)$-Planar Graphs

Understanding the maximum edge density of the $(k, p)$-planar graphs serves two purposes. First, given a graph $G$ and positive integers $k$ and $p$, we can rule out the possibility that $G$ is $(k, p)$-planar if $G$ is too dense. Second, observing how the maximum edge density increases with $k$ and $p$ gives us an idea of how quickly the classes of $(k, p)$-planar graphs increase in size and how they compare to graph classes with known maximum densities, such as the planar graphs.

In Sections 3.2 and 3.3, we prove two bounds on the edge density of $(k, p)$-planar cluster graphs, both of which are tight for certain values of $k$ and $p$. First, we prove that any $(k, p)$-planar graph with $C \geq 3$ clusters has at most

$$(3C - 6)k^2 + \frac{k(k-1)C}{2}$$

edges, a bound that is tight when $p \geq 3k$. Second, we prove that any $(k, p)$-planar graph with $C$ clusters and at least three vertices has at most

$$(kp + 3)C - 6 + \frac{k(k-1)C}{2}$$

edges, a bound that is tight when $p < k$. In the process of proving these bounds, we bound the number of intercluster and intracluster edges in a $(k, p)$-planar graph with $C$ clusters.

Our first two bounds are useful for computing the maximum number of edges in a $(k, p)$-planar graph for which the number of clusters is specified. However, we might also wish to bound the maximum number of edges in any $(k, p)$-planar graph with

$|V|$ vertices. In Sections 3.4 and 3.5, we address this question by proving that any $(k,p)$-planar cluster graph with at least three vertices and $c_i$ clusters of cardinality $i$ for $i = 1, 2, ..., k$ has at most

$$\sum_{i=2}^{k}(c_i(ip + 3 + \frac{i(i-1)}{2})) + 3c_1 - 6$$

edges. We then maximize this result over all possible clusterings by case analysis, which allows us to tightly bound the number of edges in any $(k,p)$-planar graph according to the number of vertices in the graph. The statement of this final bound requires terms introduced in Section 3.5.

## 3.1 Preliminary Definitions

In this section, we assemble the tools required for our edge bound proofs. In particular, we introduce Euler's theorem for planar graphs and two transformations that associate $(k,p)$-planar drawings with structurally similar planar graphs. We also prove a simple bound on the number of intracluster edges in a $(k,p)$-planar drawing.

Euler's theorem for planar drawings can be stated as follows.

**Theorem 7. (Euler.)** *For any planar graph $G = (V, E)$ with $|V| \geq 3$,*

$$|E| \leq 3|V| - 6.$$

Euler's theorem indicates a natural method for bounding the number of intercluster edges in a $(k,p)$-planar drawing. First, we specify a transformation that associates every $(k,p)$-planar drawing with a planar graph. Then, we demonstrate that applying our transformation to any $(k,p)$-planar drawing of a graph $G$ with $|E|$ edges would result in a nonplanar drawing. This presents a contradiction, and allows us to conclude that $G$ is not $(k,p)$-planar.

Our first transformation simplifies a cluster graph by treating each of its clusters as a single vertex. We *contract* a cluster graph $G$ by transforming each cluster $V_i$ into a vertex $v_i$, and adding an edge $(v_i, v_j)$ if the clusters $V_i$ and $V_j$ were connected by at least one intercluster edge in $G$. Figure 3.1 provides an example of this transformation.

We refer to the graph that results from this transformation as the *contracted graph* $G_C$ of $G$. If a cluster graph $G$ is $(k,p)$-planar for some $k$ and $p$, $G_C$ is planar. To

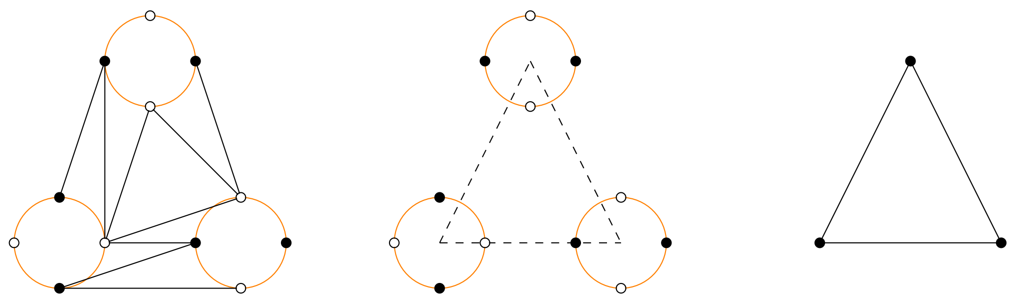


Figure 3.1: A (2,2)-planar drawing of a graph $G$ and its corresponding contracted graph $G_C$.

see this, observe that we can transform any $(k, p)$-planar drawing of $G$ into a planar drawing of $G_C$ by placing a vertex in the center of each cluster region and drawing edges on top of existing intercluster edges, which do not cross.

Our second transformation converts a $(k, p)$-planar drawing into a planar drawing by turning each port into a vertex. We *skeletonize* a $(k, p)$-planar drawing $\Gamma$ as follows. First, replace each port in $\Gamma$ with a vertex, and replace each intercluster edge in $\Gamma$ with a regular edge. Each cluster region $R_i$ in $\Gamma$ is now an empty convex space surrounded by up to $kp$ vertices. Connect these vertices in a cycle and triangulate the interior to complete the skeletonization. Figure 3.2 demonstrates the skeletonization of a $(2, 2)$-planar drawing.

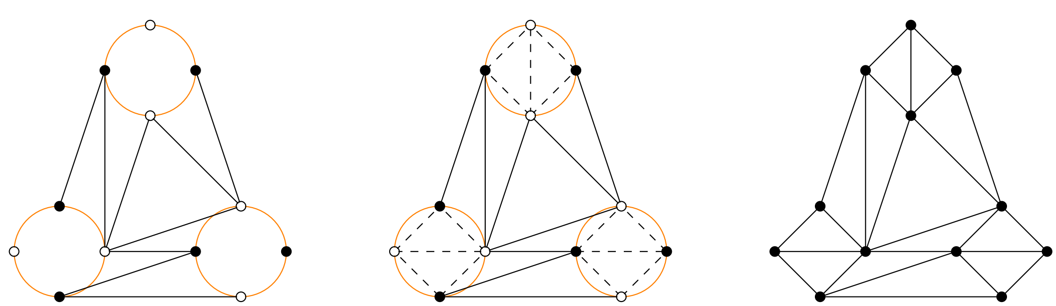


Figure 3.2: A $(2, 2)$-planar drawing $\Gamma$ and a corresponding skeleton.

We refer to the drawing resulting from a skeletonization as a *skeleton* $\Gamma_S$ of $\Gamma$. We use the indefinite article because a graph $G$ may correspond to multiple different skeletons according to differences in port order and intracluster triangulation. However, for our purposes we need not distinguish between different skeletons of the same graph. Regardless of the skeleton created, skeletonization adds no edge crossings and thus $\Gamma_S$ is planar. Finally, we note that skeletonizing a cluster region $R_i$ with $P_i$ ports creates exactly $2P_i - 3$ edges if $P_i > 1$.

Although the maximum number of intercluster edges in a $(k,p)$-planar graph depends on several factors, including the number and ordering of ports, the maximum number of intracluster edges in a $(k,p)$-planar graph is determined solely by the size of each cluster. The following bound on the number of intracluster edges in a $(k,p)$-planar drawing will be used in each of our edge bound proofs.

**Lemma 2.** *Let $G$ be a $(k,p)$-planar cluster graph with $c_i$ clusters of cardinality $i$ for $i = 0, 1, ..., k$. Letting $|E_{intracluster}|$ be the number of intracluster edges in $G$, we have that*

$$|E_{intracluster}| \leq \sum_{i=2}^{k}(c_i \frac{i(i-1)}{2}).$$

*Proof.* Let $G$ be a $(k,p)$-planar cluster graph. $G$ has no intracluster edges corresponding to single-vertex clusters. Each cluster of size $i > 1$ may have at most $\binom{i}{2} = \frac{i(i-1)}{2}$ intracluster edges. Summing the maximum number of intracluster edges over every cluster results in our bound. □

Finally, we introduce the notion of maximality for $(k,p)$-planar graphs as an extension of the notion of maximality for planar graphs. We say that a graph $G$ is *maximal $(k,p)$-planar* if the addition of any edge to $G$ results in a graph that is not $(k,p)$-planar.

## 3.2 A $p$-Independent Edge Bound Parameterized by Number of Clusters

In this section, we prove our first bound on the number of edges in a $(k,p)$-planar graph with $C$ clusters. We then show that this bound is tight when $p \geq 3k$ by demonstrating a family of maximal $(k,p)$-planar graphs that achieves our bound.

**Theorem 8.** *For any $(k, p)$-planar cluster graph $G = (V, E)$ with $C \geq 3$ clusters,*

$$|E| \leq (3C - 6)k^2 + \frac{k(k-1)C}{2}.$$

*Proof.* Let $G = (V, E)$ be a $(k, p)$-planar cluster graph with $C \geq 3$ clusters. By Lemma 2, we observe that $G$ has no more than $\frac{k(k-1)C}{2}$ intracluster edges. Thus it suffices to show that $G$ has no more than $(3C - 6)k^2$ intercluster edges to establish Theorem 8.

Let $G_C = (V_C, E_C)$ be the contracted graph of $G$. Because $G_C$ is planar and $|V_C| = C \geq 3$ by assumption, Euler's edge bound implies the following.

$$|E_C| \leq 3|V_C| - 6 = 3C - 6. \tag{3.1}$$

Because each cluster in $G$ has no more than $k$ vertices, there can be at most $k^2$ edges between any two clusters in $G$. Thus each edge of $G_C$ corresponds to no more than $k^2$ edges of $G$, and we have that

$$|E_{intercluster}| \leq |E_C|k^2. \tag{3.2}$$

Combining Equations 3.1 and 3.2, we have that

$$|E_{intercluster}| \leq (3C - 6)k^2 \tag{3.3}$$

which in combination with our bound on $|E_{intracluster}|$ establishes Theorem 8. □

We proceed to prove that the edge bound provided by Theorem 8 is tight when $p \geq 3k$. To do so, we demonstrate a family of $(k, p)$-planar cluster graphs with precisely $(3C - 6)k^2 + \frac{k(k-1)C}{2}$ edges.

**Theorem 9.** *For any pair of integers $k \geq 1$ and $C_0$, there exists a $(k, 3k)$-planar graph with $C \geq C_0$ clusters and $(3C - 6)k^2 + \frac{k(k-1)C}{2}$ edges.*

*Proof.* Given a positive integer $k$, we demonstrate a $(k, 3k)$-planar drawing $\Gamma_k$ with $C = 3$ and the maximum $(3C - 6)k^2$ intercluster edges. We then show show how $\Gamma_k$ can be repeatedly augmented to generate a $(k, 3k)$-planar drawing with $(3C - 6)k^2$

intercluster edges for arbitrarily large $C$. All of our drawings use exclusively clusters of size $k$, so we may assume that each corresponds to a $(k, p)$-planar graph $G$ with $\frac{k(k-1)C}{2}$ intracluster edges by Lemma 2. These demonstrations suffice to prove Theorem 9.

We say that two cluster regions $R_1$ and $R_2$ in a $(k, p)$-planar drawing are *fully connected* if they are connected by $k^2$ edges as shown in Figure 3.3. On the perimeter of $R_1$, $k$ ports of $R_1$ serve as the endpoints of $k^2$ edges between $R_1$ and $R_2$. These edges connect each vertex in $V_1$ to each vertex in $V_2$, which requires the use of $k(k-1)+1$ ports on the perimeter of $R_2$. We refer to the cluster region which uses $k$ ports as the *small end* of the full connection and to the cluster region which uses $k(k-1)+1$ ports as the *large end* of the full connection.

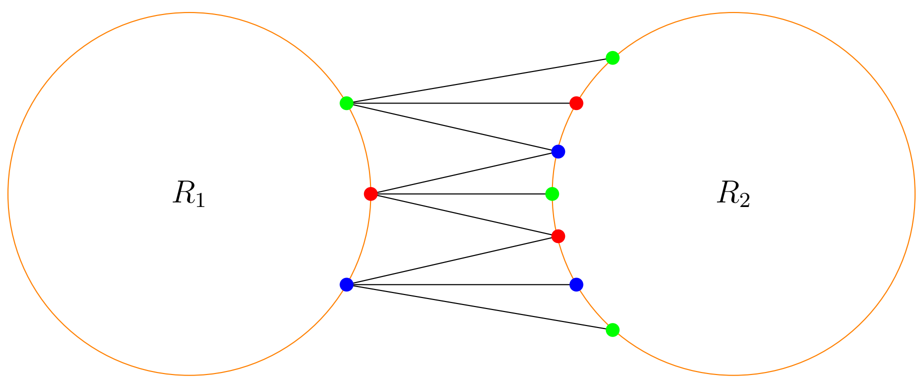



Figure 3.3: Two fully connected 3-cluster regions, $R_1$ and $R_2$.

We refer to any region in our drawing bordered by at least four ports as a *free region*. Fully connecting two clusters creates numerous regions adjacent to three ports and a large free region. We note that exactly $k-2$ ports on the small end of a full connection and $k(k-1)-1$ ports on the large end of a full connection do not border the free region.

To form the $(k, 3k)$-planar drawing $\Gamma_k$, fully connect three cluster regions so that each cluster region is the large end of one full connection and the small end of another, as shown in Figure 3.4. The resulting drawing has two free regions. Subtracting the ports rendered inaccessible by each full connection, the two free regions border a total of

$$3k \cdot k - (k-2) - (k(k-1)-1) = 2k^2 + 3$$

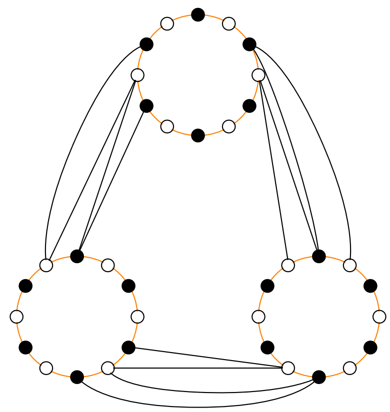


Figure 3.4: The drawing $\Gamma_2$.

ports from each cluster region. In $\Gamma_k$, we split the unused ports so that $k^2 + 1$ ports of each cluster region border both the interior and the exterior free region.

$\Gamma_k$ has 3 clusters and $3k^2$ intercluster edges. When $C = 3$,

$$3k^2 = (3C - 6)k^2,$$

so $\Gamma_k$ is a maximal $(k, 3k)$-planar drawing.

To extend our construction, nest one copy of $\Gamma_k$ inside another as illustrated in Figure 3.5. Fully connect each cluster in the inner copy of $\Gamma_k$ to two clusters in the outer copy of $\Gamma_k$, as the small end of one full connection and the large end of another full connection. Subtracting one port which can be used by both connections, this requires

$$k + (k(k - 1) + 1) - 1 = k^2$$

accessible ports, which is guaranteed by construction.

Our new drawing has three additional clusters and $9k^2$ additional edges, and thus remains maximal. Moreover, the interior of the drawing has space to embed a further $\Gamma_k$ subdrawing. Thus we can repeat our nesting operation an arbitrary number of times to generate a drawing of a maximal $(k, 3k)$-planar graph with at least $C_0$ clusters for arbitrarily large values of $C_0$. $\square$

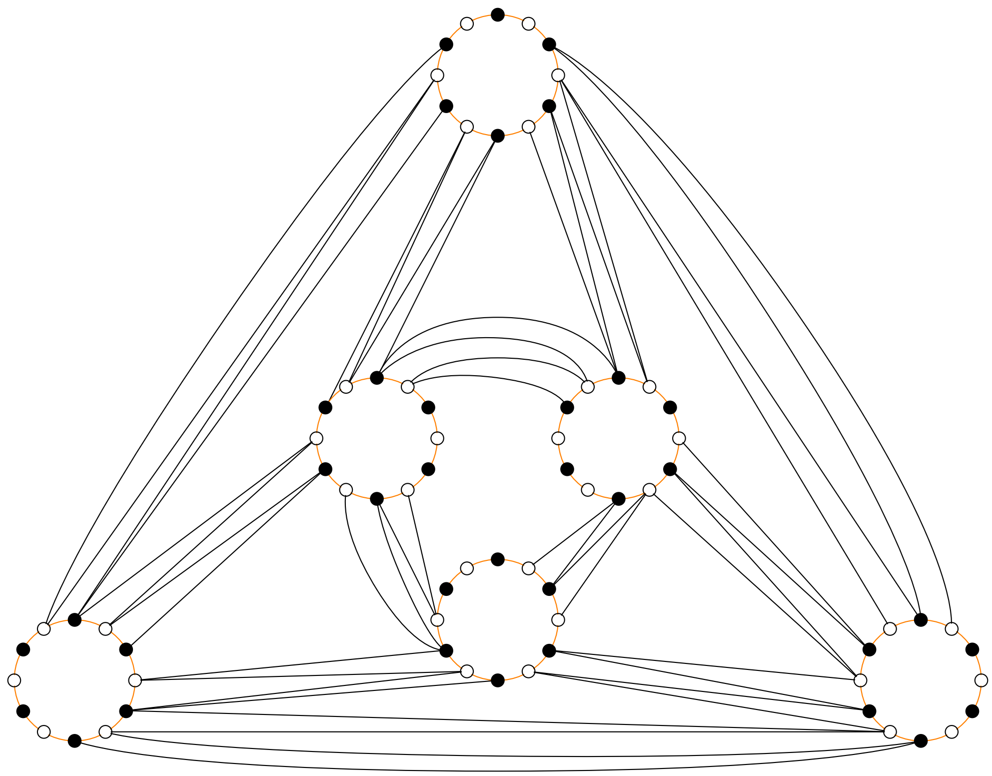


Figure 3.5: One $\Gamma_2$ subdrawing nested inside a second $\Gamma_2$ subdrawing.

## 3.3 A $p$-Dependent Edge Bound Parameterized by Number of Clusters

In this section, we prove our second bound on the number of edges in a $(k, p)$-planar graph with $C$ clusters. We then show that this bound is tight when $p < k$ by demonstrating a family of maximal $(k, p)$-planar graphs that achieves our bound.

**Theorem 10.** *For any $(k, p)$-planar cluster graph $G = (V, E)$ with $|V| \geq 3$,*

$$|E| \leq (kp + 3)C - 6 + \frac{k(k-1)C}{2}.$$

.

*Proof.* Let $G = (V, E)$ be a $(k, p)$-planar cluster graph. By Lemma 2, we observe that $G$ has no more than $\frac{k(k-1)C}{2}$ intracluster edges. Thus it suffices to show that $G$ has no more than $(kp + 3)C - 6$ intercluster edges to establish Theorem 10.

Let $\Gamma$ be a $(k, p)$-planar drawing of $G$, let $\Gamma_S$ be a skeleton of $\Gamma$, and let $G_S = (V_S, E_S)$ be the graph represented by $\Gamma_S$. Because $G_S$ is planar and $|V_S| \geq |V| \geq 3$ by assumption, Euler's edge bound implies the following.

$$|E_S| \leq 3|V_S| - 6. \tag{3.4}$$

Let $E_{intercluster}$ refer to the set of intercluster edges of $G$, and let $P_i$ be the set of ports on the perimeter of cluster region $R_i$ in $\Gamma$. $|E_S|$ is equal to $|E_{intercluster}|$ plus the number of edges added in place of each cluster region of $\Gamma$ to create $\Gamma_S$. If $|P_i| > 1$, skeletonizing $R_i$ creates $2|P_i| - 3$ additional edges, and if $|P_i| = 1$, skeletonizing $R_i$ creates $0 = 2|P_i| - 2$ additional edges. Thus, letting $c_1$ be the number of singleton clusters in $\Gamma$, we have that

$$|E_{intercluster}| + \sum_{i=1}^{C}(2|P_i| - 3) + c_1 = |E_S|. \tag{3.5}$$

By subtracting terms from the lefthand side of Equation 3.5 and substituting for $|E_S|$ according to Equation 3.4, we conclude that

$$|E_{intercluster}| \leq 3|V_S| - 6 - \sum_{i=1}^{C}(2|P_i| - 3) - c_1. \tag{3.6}$$

Substituting $\sum_{i=1}^{C} |P_i|$ with $|V_S|$ and factoring out the rest of the summation, we have

$$|E_{intercluster}| \leq |V_S| + 3C - 6 - c_1. \tag{3.7}$$

Finally, because $|V_S| \leq kpC$, we have that

$$|E_{intercluster}| \leq (kp + 3)C - 6 - c_1 \leq (kp + 3)C - 6. \tag{3.8}$$

□

Theorem 10 is intended to bound the number of edges in a $(k, p)$-planar drawing based solely on a given number $C$ of clusters, so we omit the $c_1$ term from our final result. For a more precise bound that incorporates the number of clusters of each cardinality, the reader is referred to the proof of Theorem 12 in Section 3.4.

We proceed to prove that the edge bound provided by Theorem 10 is tight when $p < k$. To do so, we demonstrate a family of $(k, p)$-planar cluster graphs with precisely $(kp + 3)C - 6 + \frac{k(k-1)C}{2}$ edges.

**Theorem 11.** *For any integers $k$, $p$ and $C_0$ such that $k > p > 0$, there exists a $(k, p)$-planar graph with $C \geq C_0$ clusters and $(kp + 3)C - 6 + \frac{k(k-1)C}{2}$ edges.*

*Proof.* Given integers $k$ and $p$ such that $k > p > 0$, we demonstrate a general $(k, p)$-planar drawing $\Gamma_{k,p}$ with $C = 3$ and the maximum $(kp+3)C-6$ intercluster edges. We then show how $\Gamma_{k,p}$ can be repeatedly augmented to generate a $(k, p)$-planar drawing with $(kp + 3)C - 6$ intercluster edges and an arbitrarily large number of clusters. All of our drawings use $k$-clusters exclusively, so we may assume that each corresponds to a graph with $\frac{k(k-1)C}{2}$ intracluster edges by Lemma 2. These demonstrations suffice to prove Theorem 11.

We say that two cluster regions $R_1$ and $R_2$ are *kp-connected* if they are connected by $kp + 1$ edges as shown in Figure 3.6. On the perimeter of $R_1$, $p + 1$ ports of $R_1$ corresponding to $k$ distinct vertices serve as the endpoints of $kp + 1$ edges. $p$ ports of $R_1$ are adjacent to $k$ edges each, and 1 additional port is adjacent to one additional edge. On the perimeter of $R_2$, $p(k - 1) + 1$ ports serve as the endpoints for edges from $R_1$. We refer to the cluster region that uses $p + 1$ ports as the *small end* of the *kp*-connection and the region that uses $p(k - 1) + 1$ ports as the *large end* of the *kp*-connection. We place ports around the perimeter of each cluster region in a consistent sequence, which ensures that any two cluster regions with $p + 1$ and $p(k - 1) + 1$ facing ports may be *kp*-connected to each other.

*kp*-connecting two clusters creates numerous regions adjacent to three ports and a large free region. We note that exactly $p-1$ ports on the small end of a *kp*-connection and $p(k - 1) - 1$ ports on the large end of a *kp*-connection do not border the free region.

To form the $(k, p)$-planar drawing $\Gamma_{k,p}$, *kp*-connect three clusters so that each is the small end of one *kp*-connection and the large end of another, as shown in Figure 3.7. Subtracting the ports rendered inaccessible by each *kp*-connection, the interior and exterior regions of $\Gamma_{k,p}$ border a total of

$$k \cdot p - (p - 1) - (p(k - 1) - 1) = 2$$

ports from each cluster region. Thus the interior and exterior regions are each adjacent

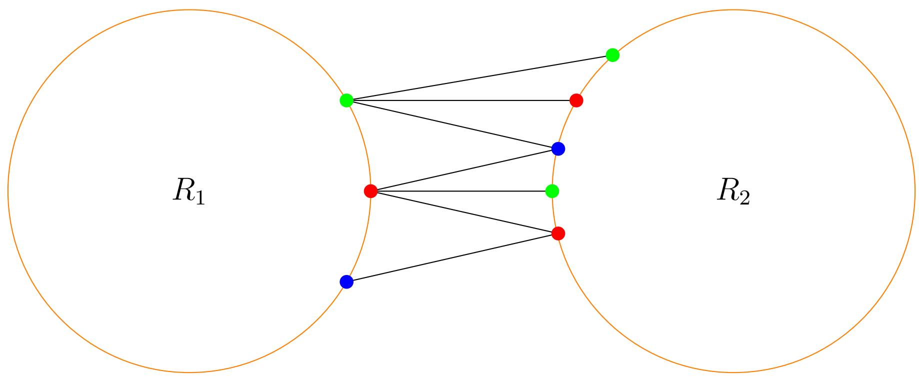



Figure 3.6: Two $kp$-connected 3-cluster regions, $R_1$ and $R_2$, with 2 ports per vertex.

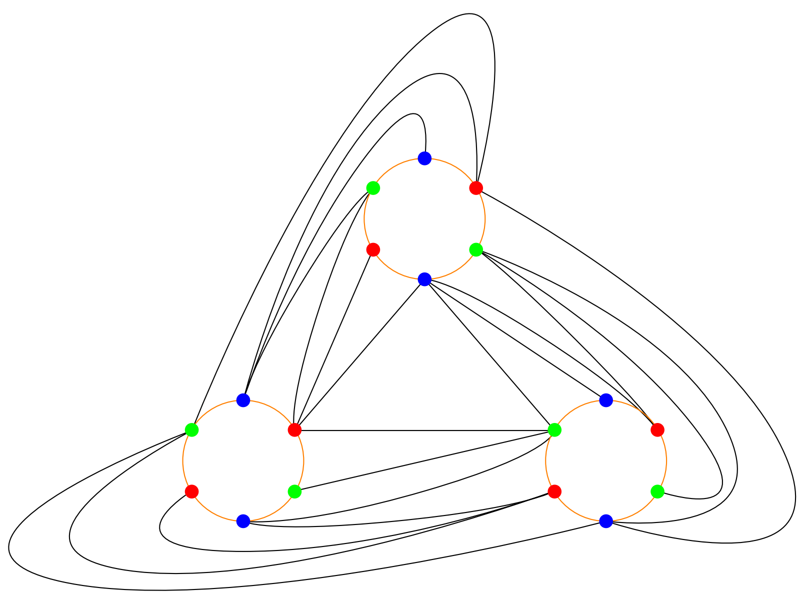


Figure 3.7: The drawing $\Gamma_{3,2}$.

to one port from each cluster.

$\Gamma_{k,p}$ has 3 clusters and $3(kp+1)$ intercluster edges. As in the $C = 3$ case,

$$3(kp+1) = (kp+3)3 - 6 = (kp+3)C - 6,$$

$\Gamma_{k,p}$ is a maximal $(k, p)$-planar drawing.

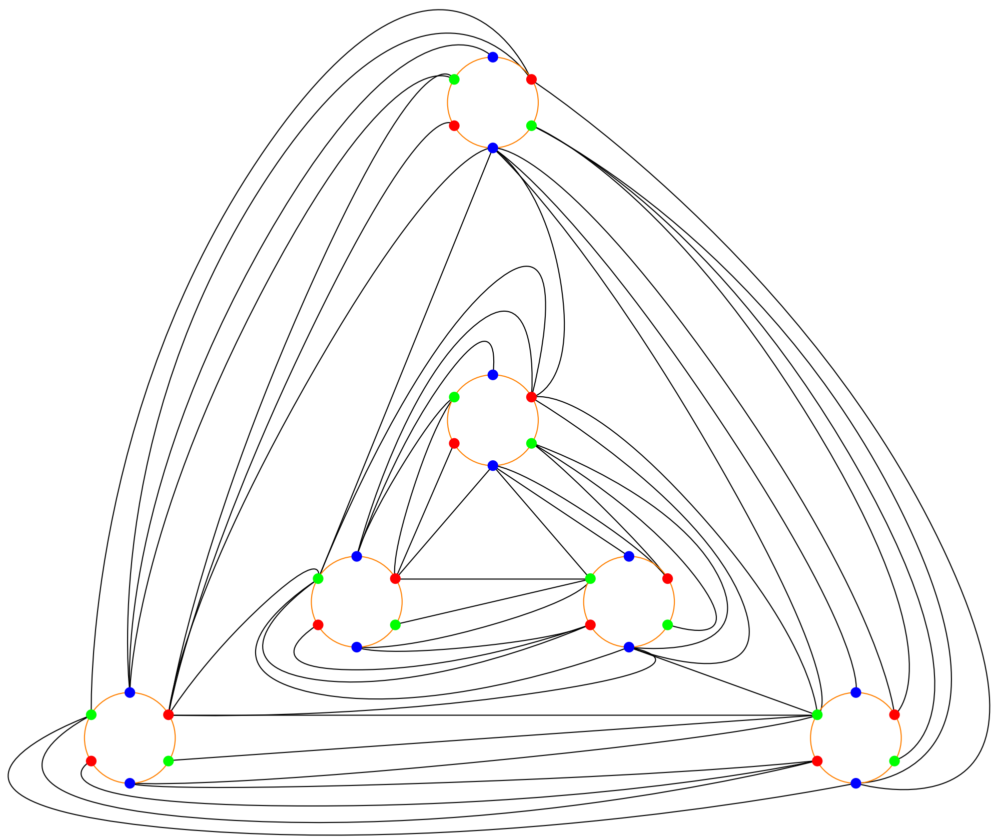


Figure 3.8: One $\Gamma_{3,2}$ subdrawing nested inside a second $\Gamma_{3,2}$ subdrawing.

To extend $\Gamma_{k,p}$, nest one copy of $\Gamma_{k,p}$ inside another as illustrated in Figure 3.8. Triangulate the six ports accessible from the interior of the outer copy of $\Gamma_{k,p}$ to create an additional six edges.

Our new drawing has three additional clusters and $3(kp+3)$ additional edges, and thus remains maximal. Moreover, the interior of our new drawing has space to embed a further $\Gamma_{k,p}$ subdrawing. Thus we can repeat our nesting operation an arbitrary number of times to generate a drawing of a maximal $(k, p)$-planar graph with at least $C$ clusters for arbitrarily large values of $C$. $\square$

## 3.4 An Edge Bound Parameterized by Clustering

In this section, we employ a method similar to the proof of Theorem 10 to establish a bound on the number of edges in a $(k, p)$-planar cluster graph that depends on the cardinality of each cluster. We can then compute the maximum bound over all possible clusterings to generate a bound parameterized by $k$, $p$, and $|V|$ alone.

The result is as follows.

**Theorem 12.** *Let $G = (V, E)$ be a $(k, p)$-planar cluster graph with $|V| \geq 3$, and let $c_i$ denote the number of clusters of $G$ with cardinality $i$ for all $i$ = 0, 1, ..., $k$. Then*

$$|E| \leq \sum_{i=2}^{k}(c_i(ip + 3 + \frac{i(i-1)}{2})) + 3c_1 - 6.$$

*Proof.* Let $G = (V, E)$ be a $(k, p)$-planar cluster graph, let $\Gamma$ be a $(k, p)$-planar drawing of $G$, let $\Gamma_S$ be a skeleton corresponding to $\Gamma$, and let $G_S = (V_S, E_S)$ be the graph corresponding to $\Gamma_S$. Finally, let $E_{intercluster}$ and $E_{intracluster}$ refer to the sets of intercluster and intracluster edges of $G$.

First, recall Equation 3.7 from Section 3.3, which states,

$$|E_{intercluster}| \leq |V_S| + 3C - 6 - c_1. \tag{3.9}$$

Upon inspection, this bound on $|E_{intercluster}|$ is maximized when $|V_S|$ is maximized, which occurs when $\Gamma_S$ corresponds to a $(k, p)$-planar drawing with exactly $p$ ports per vertex. Any $(k, p)$-planar drawing can be transformed into an equivalent $(k, p)$-planar drawing with the maximum number of ports by adding ports until every vertex is associated with exactly $p$ ports. Because this transformation changes neither $|E_{intercluster}|$ nor $|E_{intracluster}|$, we may assume without loss of generality that $\Gamma$ is a $(k, p)$-planar drawing in which each vertex is associated with exactly $p$ ports.

Thus $E_S$ consists of $E_{intercluster}$ as well as $2ip - 3$ skeleton edges created for each cluster of size $i > 1$. We have that

$$|E_{intercluster}| + \sum_{i=2}^{k} c_i(2ip - 3) = |E_S|. \tag{3.10}$$

As $|V_S| \geq |V| \geq 3$ by assumption, Euler's edge bound guarantees that $|E_S| \leq$

$3|V_S|-6$. Furthermore, as every vertex in $G$ is associated with $p$ ports in $\Gamma$, $|V_S| = \sum_{i=2}^{k}(c_i ip) + c_1$. Combining these results with Equation 3.10, we have that

$$|E_{intercluster}| + \sum_{i=2}^{k} c_i(2ip-3) \leq 3(\sum_{i=2}^{k}(c_i ip) + c_1) - 6 \tag{3.11}$$

which reduces to

$$|E_{intercluster}| \leq \sum_{i=2}^{k}(c_i ip + 3c_i) + 3c_1 - 6. \tag{3.12}$$

By Lemma 2, we have that

$$|E_{intracluster}| \leq \sum_{i=2}^{k}(c_i \frac{i(i-1)}{2}). \tag{3.13}$$

As $|E| = |E_{intercluster}| + |E_{intracluster}|$, summing Equations 3.12 and 3.13 yields

$$|E| \leq \sum_{i=2}^{k}(c_i(ip + 3 + \frac{i(i-1)}{2})) + 3c_1 - 6, \tag{3.14}$$

which completes the proof. □

Theorem 12, like Theorem 10, computes an upper bound on the number of intercluster edges by counting the number of edges that would be created by triangulating the ports of a $(k,p)$-planar drawing in intercluster space. Such a triangulation is always possible in the $p=1$ case, in which every port corresponds to a distinct vertex. Thus the bound provided by Theorem 12 is tight, at least for $p=1$.

## 3.5 An Edge Bound Parameterized by Number of Vertices

Theorem 12 applies to $(k,p)$-planar graphs with a fixed clustering. In this section, we prove an edge bound for $(k,p)$-planar graphs that depends solely on $k$, $p$, and $|V|$ by maximizing Theorem 12 over every possible clustering.

Given a $(k,p)$-planar graph $G=(V,E)$ with $|V| \geq 3$, we can view Theorem 12 as a function that takes as input a partition $P$ of $V$ and returns a bound on the maximum number of edges in a $(k,p)$-planar drawing of $G$ according to $P$. For the sake of simplicity, in this section we treat vertices as identical elements and treat partitions with the same number of clusters of each cardinality as equivalent. Using these conventions, we can bound the maximum number of edges in any $(k, p)$-planar drawing of $G$ by maximizing the function

$$f(P) = \sum_{i=2}^{k}(c_i(ip + 3 + \frac{i(i-1)}{2})) + 3c_1 \tag{3.15}$$

over all partitions $P$ of $V$, where $c_i$ denotes the number of clusters of cardinality $i$ in $P$. $f(P)$ is identical to the right side of the bound provided by Theorem 12 except for the omission of the constant term -6.

The function $f$ can be conceived of as a sum that increases by a certain number for each vertex in $V$. Thus each vertex in a cluster of cardinality 1 increases $f(P)$ by 3, and each vertex in a cluster of cardinality $i$ increases $f(P)$ by

$$\frac{ip + 3 + \frac{i(i-1)}{2}}{i} = \frac{i-1}{2} + p + \frac{3}{i}.$$

We formalize this statement of the problem by defining the following function that tracks the contribution of each vertex in $V$ to $f$ according to $P$.

**Definition 7.** *For $p \in \mathbb{Z}^+$, the* edge efficiency function $\eta_p : \mathbb{Z}^+ \to \mathbb{R}$ *is defined*

$$\eta_p(i) = \begin{cases} 3 & \text{if } i = 1 \\ \frac{i-1}{2} + p + \frac{3}{i} & \text{otherwise.} \end{cases} \tag{3.16}$$

By construction, the function $f$ can be restated as the sum of the edge efficiency function over all vertices. Letting $V(v_i)$ denote the cluster corresponding to the vertex $v_i$ in $P$, we have

$$f(P) = \sum_{i=1}^{|V|} \eta_p(|V(v_i)|). \tag{3.17}$$

For every positive integer $p$, the edge efficiency function $\eta_p$ is non-decreasing, indicating that larger values of $f$ correspond to partitions $P$ composed of larger clusters. In particular, we note that $\eta_p(1) \leq \eta_p(2) = \eta_p(3)$ and that $\eta_p(i)$ increases monotonically over the integers greater than two.

The following technical lemma is necessary for our edge bound proof.

**Lemma 3.** *Let $G = (V, E)$ be a graph. Then there exists a partition $P^*$ on $V$ with*

$$|P^*| = \left\lceil \frac{|V|}{k} \right\rceil$$

*that maximizes the function $f$.*

*Proof.* Let $G = (V, E)$ be a graph. Assume for contradiction that no partition of cardinality $\left\lceil \frac{|V|}{k} \right\rceil$ maximizes $f$. Thus there exists a partition $P$ of $V$, with $|P| > \left\lceil \frac{|V|}{k} \right\rceil$, such that $f(P) > f(Q)$ for all partitions $Q$ with $|Q| < |P|$.

Because $|P| > \left\lceil \frac{|V|}{k} \right\rceil$, we can identify a cluster $V_j \in P$ with minimal cardinality and distribute the vertices of $V_j$ among other clusters in $P$ to create a partition $P_0$ on $V$ with $|P_0| = |P| - 1$. $P_0$ and $P$ contain the same vertices, but $P_0$ assigns each vertex to a cluster of the same size or larger than the cluster to which it is assigned by $P$.

As the edge efficiency function $\eta_p(i)$ is non-decreasing, we have that $f(P_0) \geq f(P)$ by Equation 3.17, which contradicts our assumption. Thus some partition $P^*$ on $V$ with $|P^*| = \left\lceil \frac{|V|}{k} \right\rceil$ maximizes $f$. □

.

These observations allow us to prove the following general edge bound.

**Theorem 13.** *Let $G = (V, E)$ be a graph with $|V| \geq 3$, and let $q$ and $r$ be integers such that $|V| = qk + r$, $0 \leq r < k$. If $r \neq 1$ or $p + 2 \leq k$,*

$$|E| \leq qk\,\eta_p(k) + r\,\eta_p(r) - 6,$$

*and if $r = 1$ and $p + 2 > k$,*

$$|E| \leq (q-1)k\,\eta_p(k) + (k-1)\,\eta_p(k-1) + 2\,\eta_p(2) - 6.$$

*Proof.* Let $G = (V, E)$ be a $(k, p)$-planar graph with $|V| \geq 3$, and let $q$ and $r$ be integers such that $|V| = qk + r$, $0 \leq r < k$, determined by the division algorithm. Let $P^*$ be a partition of $V$ with $|P^*| = \left\lceil \frac{|V|}{k} \right\rceil$ that maximizes $f$. Let $Q = \{V_1, V_2, ..., V_{q+1}\}$ be the partition of $V$ with $|V_i| = k$ for $i = 1, 2, ..., q$ and $|V_{q+1}| = r$. If $r = 0$, remove the empty set $V_{q+1}$ from $Q$.

To establish the first equation of Theorem 13, we show $P^* = Q$ when $r \neq 1$. The second equation of Theorem 13 results from an exception to our proof in the $r = 1$ case.

*Case 1.* Suppose $r = 0$. In this case, $|P^*| = \left\lceil \frac{|V|}{k} \right\rceil = |V|/k = |Q|$. When $k$ divides $|V|$, there is exactly one partition of cardinality $\frac{|V|}{k}$, so $P^* = Q$.

*Case 2.* Suppose $r > 1$ and assume for contradiction that $P^* \neq Q$. Let $V_{min}$ be a cluster of $P^*$ with minimal cardinality. Because $P^* \neq Q$ by assumption, $P^*$ contains a second cluster $V_n$ such that $|V_{min}| \leq |V_n| < k$.

Assume for contradiction that $|V_{min}| \leq 2$. If $|V_{min}| = 1$, then the single vertex in $V_{min}$ could be added to $V_n$ to create a partition with fewer than $|P^*|$ clusters. This is contradictory, as $|P^*|$ is minimal by construction.

Suppose instead that $|V_{min}| = 2$. If $|V_n| < k - 1$, or if there existed a second partition $V_m$ with $|V_m| < k$, then the vertices of $V_{min}$ could again be distributed among other clusters, causing a contradiction by creating a partition smaller than $|P^*|$. As $|V_n| < k$, we are left to conclude that that $V_n$ is the only cluster in $P^*$ apart from $V_{min}$ with cardinality less than $k$ and that $|V_n| = k - 1$. However, this implies that $r = 1$ and contradicts our case hypothesis. Thus $|V_{min}| > 2$.

Let $P_0^*$ be the partition of $V$ created by moving a vertex from $V_{min}$ to $V_n$. The addition of the vertex to $V_n$ increases $f(P_0^*)$ relative to $f(P^*)$, while the removal of the vertex from $V_{min}$ decreases $f(P_0^*)$ relative to $f(P^*)$. In particular, the addition of the vertex to $V_n$ increases $f$ by

$$((|V_n| + 1)p + 3 + \frac{|V_n + 1||V_n|}{2}) - (|V_n|p + 3 + \frac{|V_n||V_n - 1|}{2}) = p + |V_n|,$$

and the removal of the vertex from $V_{min}$ decreases $f$ by

$$(|V_{min}|p+3+\frac{|V_{min}||V_{min} - 1|}{2})-((|V_{min}|-1)p+3+\frac{|V_{min} - 1||V_{min} - 2|}{2}) = p+|V_{min}|-1.$$

In total, moving the vertex from $V_{min}$ to $V_n$ increases $f(P_0^*)$ by

$$(p + |V_n|) - (p + |V_{min}| - 1) = |V_n| - |V_{min}| + 1 \geq 1$$

relative to $f(P^*)$. This is a contradiction, and thus $P^* = Q$ in the $r > 1$ case.

*Case 3.* Suppose $r = 1$. In this case, our argument is identical to Case 2 except for the possibility that $|V_{min}| = 2$, in which case $|V_n| = k - 1$. Consider $P_0^*$, the partition of $V$ created by moving a vertex from $V_{min}$ to $V_n$, when $|V_{min}| = 2$. Adding a vertex to $V_n$ increases $f$ by $p + |V_n| = p + k - 1$, but removing a vertex from $|V_{min}|$ decreases $f$ by

$$(|V_{min}|p + 3 + \frac{|V_{min}||V_{min} - 1|}{2}) - 3 = 2p + 1.$$

Thus moving a vertex from $V_{min}$ to $V_n$ results in a net change of $(p+k-1)-(2p+1) = k - (p + 2)$. Thus when $r = 1$ and $p + 2 > k$, $f(P)$ is maximized by the clustering in which all clusters have size $k$ except for clusters $V_n$ and $V_{min}$ with $|V_n| = k - 1$, $|V_{min}| = 2$.

Case 3 accounts for the caveat in Theorem 13. However, unless $r = 1$ and $p+2 > k$, we have that $|E| \le qk\,\eta_p(k) + r\,\eta_p(r) - 6$ according to the maximal partition $Q$. $\square$

Because Theorem 13 reflects a specialized case of Theorem 12, the bound provided by Theorem 13 is likewise tight in the $p = 1$ case.

## 3.6 Conclusion

In this chapter, we first proved two bounds on the maximum edge density of $(k,p)$-planar cluster graphs with $C$ clusters. The first bound, which depends on the maximum number of ports per vertex, becomes tight when the number of ports is limited. The second bound, which is independent of the maximum number of ports per vertex, becomes tight when enough ports are allowed to fully connect any two clusters. The problem of finding a tight edge bound in the $k \le p < 3k$ case remains open.

In the second part of the chapter, we proved a bound on the maximum edge density of $(k,p)$-planar cluster graphs with fully specified clusterings. Then, we analyzed this bound over all possible clusterings to generate a bound parameterized by $k$, $p$, and $|V|$ alone. Both bounds are tight in the $p = 1$ case.

# Chapter 4

# Hardness of Deciding $(k, p)$-Planarity

In this chapter, we address the hardness of the $(k, p)$-planarity decision problem: given positive integers $k$ and $p$ and a graph $G$, does $G$ admit a clustering such that the resulting cluster graph is $(k, p)$-planar?

In Section 4.1, we consider the hardness of deciding $(k, 1)$-planarity. When $k \leq 4$, the hardness of deciding $(k, 1)$-planarity follows from the results proved in Chapter 2. In addition, we consider a modified version of the $(k, 1)$-planarity decision problem in which a fixed clustering is specified. We prove that when a clustering is fixed, $(k, 1)$-planarity is decidable in linear time for all positive integers $k$.

In Section 4.2, we prove that deciding $(2, 2)$-planarity is NP-complete by reduction from Planar Monotone 3-SAT.

## 4.1 Hardness of Deciding $(k, 1)$-Planarity

In Section 2.2.1, we proved that the classes of $(1, 1)$-planar graphs, $(2, 1)$-planar graphs, and $(3, 1)$-planar graphs are equivalent to the class of planar graphs. The planarity of a graph can be tested in linear time [19], so $(k, 1)$-planarity can be tested in linear time for $k \leq 3$.

In Section 2.2.2, we proved that the $(4, 1)$-planar graphs are exactly the IC-planar graphs. As testing IC-planarity is NP-complete [7], the $(4, 1)$-planarity decision problem is NP-complete. The hardness of testing $(k, 1)$-planarity for $k > 4$ remains an open problem.

The problem of deciding $(k, 1)$-planarity for a graph $G$ with a fixed clustering is significantly easier than the general case. In fact, the $(k, 1)$-planarity decision problem for cluster graphs can be reduced to the problem of planarity testing.

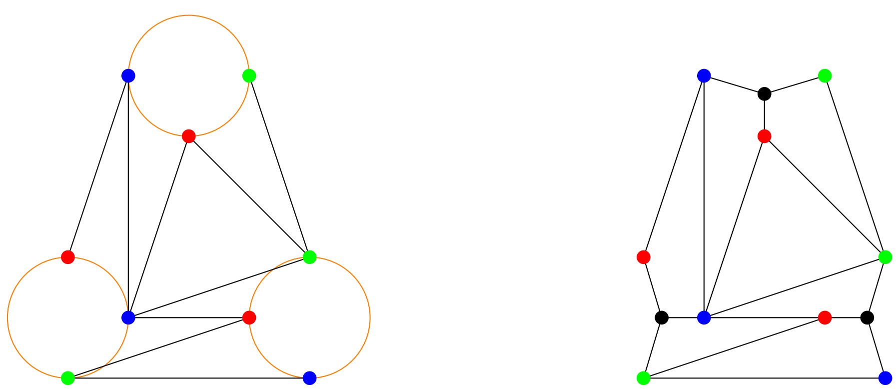
**Theorem 14.** *Let $G = (V, E)$ be a graph and let $P$ be a partition of $V$. Given a positive integer $k$, we can determine whether or not $G$ admits a $(k, 1)$-planar drawing clustered according to $P$ in linear time.*

*Proof.* Let $G = (V, E)$ be a graph and let $P$ be a partition of $V$. If any cluster in $P$ has cardinality greater than $k$, we can reject $G$ immediately. We proceed with the assumption that every cluster in $P$ has cardinality at most $k$.

We construct a graph $G'$ that is planar if and only if $G$ is $(k, 1)$-planar. Because $G'$ can be constructed in linear time, this suffices to prove Theorem 14.

Construct $G'$ from $G$ as follows. Add a vertex $u_j$ to $G$ for every cluster $V_j \in P$. Then, for each vertex $v \in V_j$, add the edge $(v, u_j)$. Every cluster $V_i \in P$ is thus represented by a star subgraph $S_i \subset G'$.

We proceed to prove that that $G$ is $(k, 1)$-planar if and only if $G'$ is planar.



(a) A $(3, 1)$-planar drawing of a graph $G$. (b) A planar drawing of a graph $G'$.

Figure 4.1: Drawings of $G$ and corresponding graph $G'$.

First, suppose that $G$ is $(k, 1)$-planar and let $\Gamma$ be a $(k, 1)$-planar drawing of $G$. $\Gamma$ can be transformed into a planar drawing of $G'$ by placing a vertex $u_i$ inside each cluster region $R_i$, connecting $u_i$ to each port of $R_i$, and removing the cluster boundary.

This process transforms the $(k, 1)$-planar drawing illustrated in Figure 4.1a into the planar drawing illustrated in Figure 4.1b.

Likewise, suppose that $G'$ is planar and let $\Gamma'$ be a planar drawing of $G'$. $\Gamma'$ can be transformed into a $(k, 1)$-planar drawing of $G$ by tracing the perimeter of a cluster region $R_i$ around the spokes of $S_i$ as tightly as necessary to ensure that no intercluster edges intersect $R_i$. Remove each vertex $u_i$ and edges adjacent to $u_i$. The result is a $(k, 1)$-planar drawing $\Gamma$ of $G$. This process transforms the planar drawing illustrated in Figure 4.1b into the $(k, 1)$-planar drawing illustrated in Figure 4.1a.

As $G$ is planar if and only if $G'$ is planar, we can test the $(k, 1)$-planarity of $G$ according to $P$ in linear time by constructing $G'$ and performing a planarity test. $\square$

## 4.2 Hardness of Deciding $(2, 2)$-Planarity

In this section, we present a proof that the $(2, 2)$-planarity decision problem, or $(2, 2)$-Planarity, is NP-complete. The proof is by reduction from Planar Monotone 3-SAT, a problem shown to be NP-complete by de Berg and Khosravi [5].

A few definitions are necessary for the reduction. First, an instance of 3-SAT consists of a boolean expression $F$, where $F$ is in conjunctive normal form and each clause in $F$ contains exactly three literals. We say that an instance of 3-SAT is *monotone* if every clause consists solely of non-negated or solely of negated literals, and refer to such clauses as *positive* and *negative* clauses, respectively. We say that a 3-SAT instance with a set of variables $X$ and a set of clauses $C$ is *planar* if the graph $G$ with vertex set $X \cup C$ and edges between every clause and its constituent variables is planar.

A *rectilinear representation* of a planar 3-SAT instance is a drawing of $G$ in which each variable and clause is represented by a rectangle, all the variable rectangles are drawn along a horizontal line, the edges connecting variables and clauses are represented by vertical line segments, and the whole drawing is crossing free. Knuth and Raghunatan [22] showed that every graph corresponding to a planar 3-SAT instance has a rectilinear representation. A *monotone rectilinear representation* is a rectilinear representation of a graph G corresponding to a monotone instance of planar 3-SAT. Additionally, in a monotone rectilinear representation, all positive clauses are drawn above the line of variables and all negative clauses are drawn below the line of variables.

Figure 4.2 provides an example of a rectilinear representation. Note that in every rectilinear representation, the variables can be connected by a planar cycle. Moreover, in a monotone rectilinear representation, this cycle separates the positive clause rectangles from the negative clause rectangles.

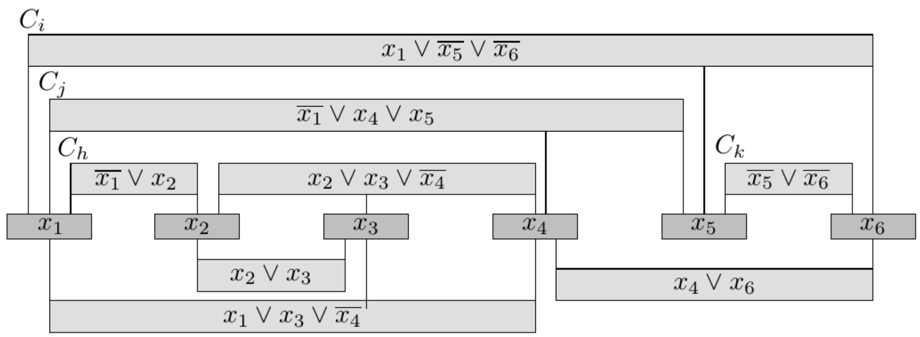



Figure 4.2: A rectilinear representation of a planar 3-SAT instance reproduced from [5].

In [5], de Berg and Khosravi show that given a monotone rectilinear representation $X$ corresponding to an instance $F$ of 3-SAT, it is NP-complete to determine if $F$ has a satisfying assignment. We refer to this decision problem as Planar Monotone 3-SAT. Because Planar Monotone 3-SAT is NP-complete, reducing $(2, 2)$-Planarity to this problem demonstrates that $(2, 2)$-Planarity is NP-hard. The following lemma is necessary for the proof.

We refer to the graph created by removing two adjacent edges from the complete graph $K_8$ as $K_{8-}$, and to the single vertex of $K_{8-}$ with degree 5 as a *K-vertex*. We use the phrase *adding a K-vertex* to a graph $G$ to refer to the operation of adding a $K_{8-}$ subgraph and associated K-vertex. For example, given a graph $G = (V, E)$ and a vertex $v \in V$, the instruction "Add a K-vertex $u$ and an edge $(u, v)$ to $G$" refers to the process of adding a $K_{8-}$ subgraph to $G$, identifying the K-vertex of this subgraph as $u$, and adding the edge $(u, v)$ to $E$.

We prove the following property of any $(2, 2)$-planar drawing that includes a $K_{8-}$ subgraph.

**Lemma 4.** *Let $K = (V_K, E_K)$ be a $K_{8-}$ subgraph of a graph $G$. In any $(2, 2)$-planar drawing of $G$, the K-vertex $v$ of $K$ is clustered with a vertex in $V_K$.*

*Proof.* Let $K = (V_K, E_K)$ be a $K_{8-}$ subgraph of a graph $G$ with K-vertex $v$. Assume for contradiction that there exists a $(2, 2)$-planar drawing $\Gamma$ of $G$ in which $v$ is not clustered with a vertex in $V_K$.

Suppose that the remaining seven vertices of $K_{8-}$ are partitioned into at least five clusters. In this case, the contracted graph of $G$ includes a $K_5$ subgraph, which contradicts our assumption that $\Gamma$ is $(2, 2)$-planar.

Alternatively, suppose that the remaining seven vertices of $K_{8-}$ are partitioned into four clusters. In this case, $E_K$ contains three 2-clusters, $v$, and an additional vertex $w$. If $v$ and $w$ are not clustered with vertices outside of $K$, $\Gamma$ contains a $(2, 2)$-planar drawing of $K$ and Theorem 12 implies that $|E_K| \leq 24$. This creates a contradiction as $|E_K| = 26$.

However, $\Gamma$ is no more plausible if $v$ or $w$ is clustered with a vertex outside of $E_K$. As $\Gamma$ is $(2, 2)$-planar by assumption, contracting the cluster regions associated with $v$ and $w$ creates a subdrawing of $\Gamma$ equivalent to a $(2, 2)$-planar drawing of $K$ with three 2-clusters and two 1-clusters. As previously observed, such a drawing is impossible.

As the remaining seven vertices of $K_{8-}$ cannot be partitioned into fewer than four clusters, our assumption that $\Gamma$ is planar creates a contradiction in every case. Thus $v$ is clustered with a vertex in $V_K$ in any $(2, 2)$-planar drawing of $G$. □

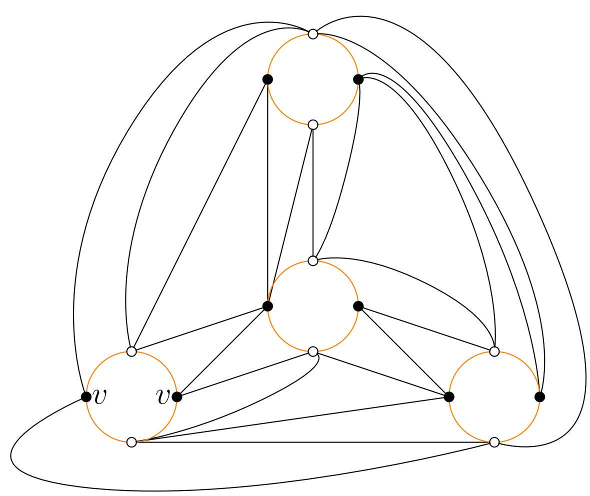



Figure 4.3: A $(2, 2)$-planar drawing of a $K_{8-}$ subgraph and K-vertex $v$.

A $(2, 2)$-planar drawing of $K_{8-}$ is possible when the vertices of $K_{8-}$ are partitioned into four clusters as illustrated in Figure 4.3. Lemma 4 ensures that if we add a K-vertex $v$ to a graph $G = (V, E)$, no $(2, 2)$-planar drawing of $G$ clusters $v$ with a vertex in $V$. Noting this useful property of the K-vertex, we proceed to establish the NP-completeness of $(2, 2)$-planarity.

**Theorem 15.** *(2,2)-Planarity is NP-complete.*

(2,2)-Planarity is trivially in NP, as an input graph $G$ is certified by a (2,2)-planar drawing.

We show that given an instance $X$ of Planar Monotone 3-SAT, we can construct in polynomial time a graph $G$, proportional to $X$ in size, that is a YES instance of (2,2)-Planarity if and only if $X$ is a YES instance of Planar Monotone 3-SAT. This suffices to prove the NP-hardness of (2,2)-Planarity.

### 4.2.1 Construction of G

For convenience, the figures in this section depict the construction of the graph $G_0$ corresponding to the Planar Monotone 3-SAT instance $X_0$ illustrated in Figure 4.4. $X_0$ corresponds to the boolean formula $F_0 = (v_1 \vee v_2 \vee v_3) \wedge (v_1 \vee v_3 \vee v_4) \wedge (\bar{v_2} \vee \bar{v_3} \vee \bar{v_4}) \wedge (\bar{v_1} \vee \bar{v_2} \vee \bar{v_4})$. In subsequent figures, we represent K-vertices and their associated $K_{8-}$ subgraphs with solid dots and represent ordinary vertices with hollow dots.

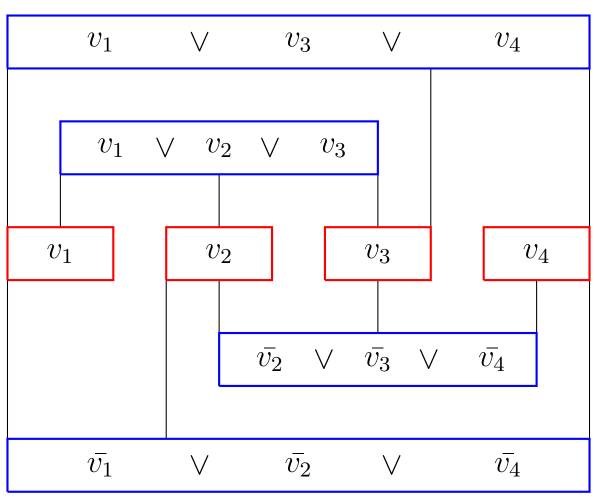



Figure 4.4: A planar monotone representation of $X_0$.

Given an instance of Planar Monotone 3-SAT $X$ corresponding to a boolean formula $F$, we construct $G$ as follows.

**The Variable Cycle**

First, add a K-vertex $v_i$ to $G$ for each variable in $F$. For convenience, we use the symbol $v_i$ to refer to both the variable in $F$ and the corresponding vertex in $G$. Connect these vertices to form a cycle in the order implied by the monotone rectilinear representation $X$. Split each edge $(v_i,\ v_{i+1})$ of the cycle by inserting the K-vertex $c_{i,i+1}$. Split the edge $(v_1,\ v_n)$ twice, adding the K-vertices $c_{0,1}$ and $c_{n,n+1}$. Each K-vertex $v_i$ is now adjacent to the vertices $c_{i-1,i}$ and $c_{i,i+1}$. Finally, duplicate the edge $(c_{0,1}, c_{n,n+1})$ and split the duplicate edges with the special K-vertices *plus* and *minus*. This construction, which we refer to as the *variable cycle*, will separate positive clause gadgets from negative clause gadgets in drawings of $G$. The variable cycle is illustrated in Figure 4.5.

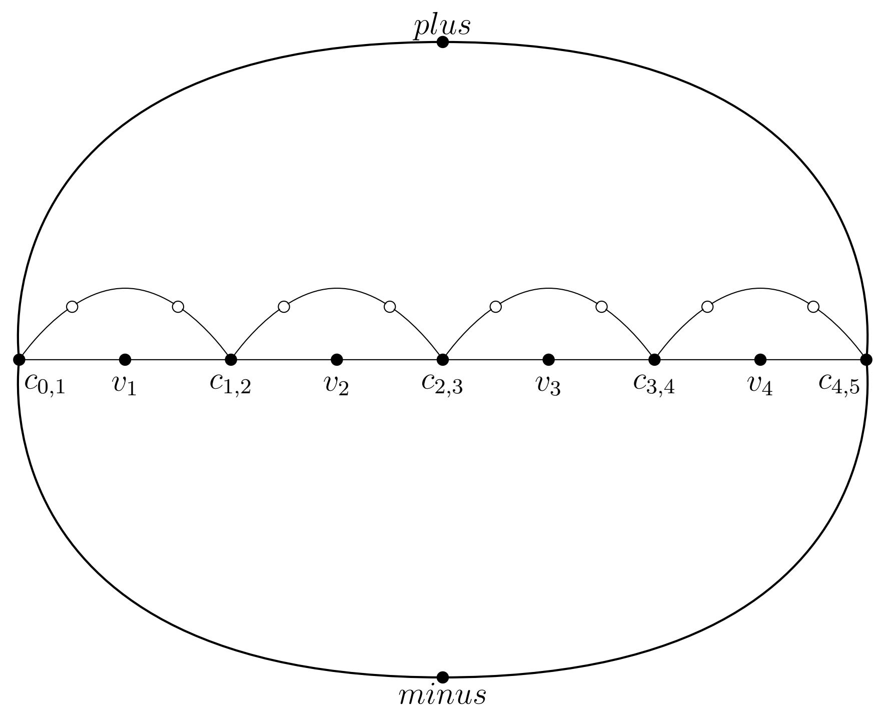
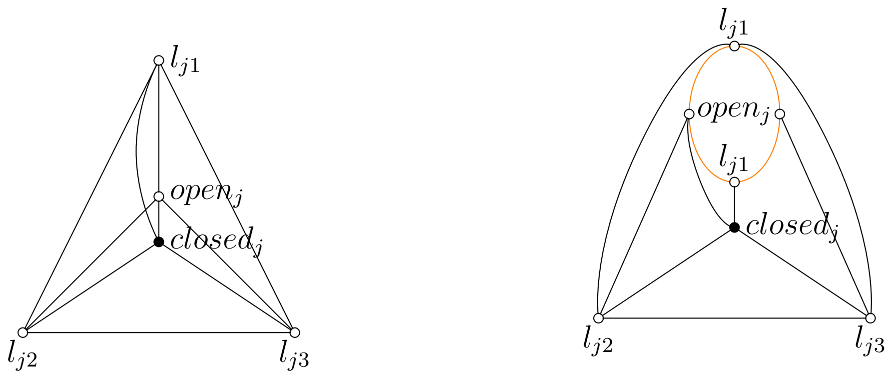



Figure 4.5: The variable cycle of $G_0$ with false literal boundaries.

The next step in the construction of $G_0$ is to augment the variable cycle with paths that we will refer to as *false literal boundaries.* Given a variable $v_i$, we let $p_i$ be the number of positive clauses and $n_i$ be the number of negative clauses of $F$ in which $v_i$ appears. Construct the false literal boundary of each vertex $v_i$ by connecting $c_{i-1,i}$ to $c_{i,i+1}$ with a path of length $max(p_i, n_i)$ as illustrated in Figure 4.5.

### The Clause Gadget

For each clause $C_j = (l_{j1} \vee l_{j2} \vee l_{j3})$ in $F$, we create a corresponding clause gadget in $G$ as follows. First, add the vertices $l_{j1}$, $l_{j2}$, $l_{j3}$, and $open_j$, and the K-vertex $closed_j$ to $G$. Create an edge between every pair of vertices as illustrated in Figure 4.6a.




(a) A clause gadget $C_j$.

(b) A $(2, 2)$-planar drawing of the clause gadget $C_j$.

Figure 4.6: Node-link and $(2, 2)$-planar drawings of the clause gadget $C_j$.

The clause gadget resembles a $K_5$ subgraph, which limits the ways in which it can be represented by a $(2, 2)$-planar drawing. We prove the following property of any $(2, 2)$-planar drawing that includes a clause gadget subgraph.

**Lemma 5.** *Let $G = (V, E)$ be a graph that includes a clause gadget subgraph containing the vertices $l_{j1}, l_{j2}, l_{j3}$, $open_j$, and $closed_j$. In any $(2, 2)$-planar drawing of $G$, two of the four vertices $l_{j1}, l_{j2}, l_{j3}$, and $open_j$ must be clustered together.*

*Proof.* First, observe that in any $(2, 2)$-planar drawing of a clause gadget, $closed_j$ must be clustered with a vertex in its associated $K_{8-}$ subgraph according to Lemma 4.

Suppose for contradiction that $G$ admits a $(2, 2)$-planar drawing $\Gamma$ in which none of $l_{j1}, l_{j2}, l_{j3}$, and $open_j$ are clustered together. In this case, the contracted graph $G_C$ of $G$ includes a $K_5$ minor. As $G_C$ is planar when $G$ is $(k, p)$-planar, this is a contradiction, which suffices to establish our lemma. □

In particular, Lemma 5 disallows the possibility of any $(2, 2)$-planar drawing of G in which $l_{j1}, l_{j2}$, and $l_{j3}$ are all clustered with vertices outside the clause gadget. However, we note that any partition that clusters a literal vertex with $open_j$ does allow a $(2, 2)$-planar drawing, as illustrated in Figure 4.6b.

#### Connecting the Clause Gadgets

We connect the clause gadgets with edges according to the relative position of clause rectangles in $X$. Roughly speaking, each clause gadget is connected by an edge to the clause gadgets above and below it in $X$, creating a forest of clause gadgets. The following procedure makes this construction precise.

For each clause rectangle $C_j$ in $X$, perform the following procedure. Let $v_1$, $v_2$ and $v_3$ denote the variable rectangles connected to $C_j$ in $X$, from left to right. By $l_{j1}$, $l_{j2}$, and $l_{j3}$ we denote the literals of $C_j$ corresponding to $v_1$, $v_2$ and $v_3$, respectively. If there exists a clause rectangle $C_k$ such that $C_j$ is nested immediately underneath $C_k$ in $X$, let $u_1$ and $u_2$ denote the variable rectangles connected to $C_k$ in $X$ on either side of $C_j$, and let $l_{k1}$ and $l_{k2}$ denote the literals of $C_k$ corresponding to $u_1$ and $u_2$, respectively. Split the edges $(l_{j1}, l_{j3})$ and $(l_{k1}, l_{k2})$ with two K-vertices and connect the new K-vertices with an edge.

If $C_j$ is not nested underneath another clause rectangle in $X$ and $C_j$ corresponds to a positive clause, split $(l_{j1}, l_{j3})$ with a K-vertex and connect the new vertex to *plus*. If $C_j$ is not nested underneath another clause rectangle in $X$ and $C_j$ corresponds to a negative clause, split $(l_{j1}, l_{j3})$ with a K- vertex and connect the new vertex to *minus*.

Each clause gadget is thus connected by an edge to the clause gadgets corresponding to neighboring clause rectangles in $X$. The newly added edges, hereafter referred to as *tree structure edges*, organize the clause gadgets into two trees, rooted at *plus* and *minus*. Figure 4.7 illustrates tree structure edges connecting the clause gadgets of the graph $G_0$.

### 4.2.2 If $X$ Is a YES Instance of Planar Monotone 3-SAT, $G$ Is a YES Instance of (2,2)-Planarity

Let $X$ be a YES instance of Planar Monotone 3-SAT, and $A$ be an assignment function satisfying $F$. We show that the graph $G$ corresponding to $X$ is $(2, 2)$-planar by constructing a $(2, 2)$-planar drawing of $G$ as follows. Replace each variable box in $X$

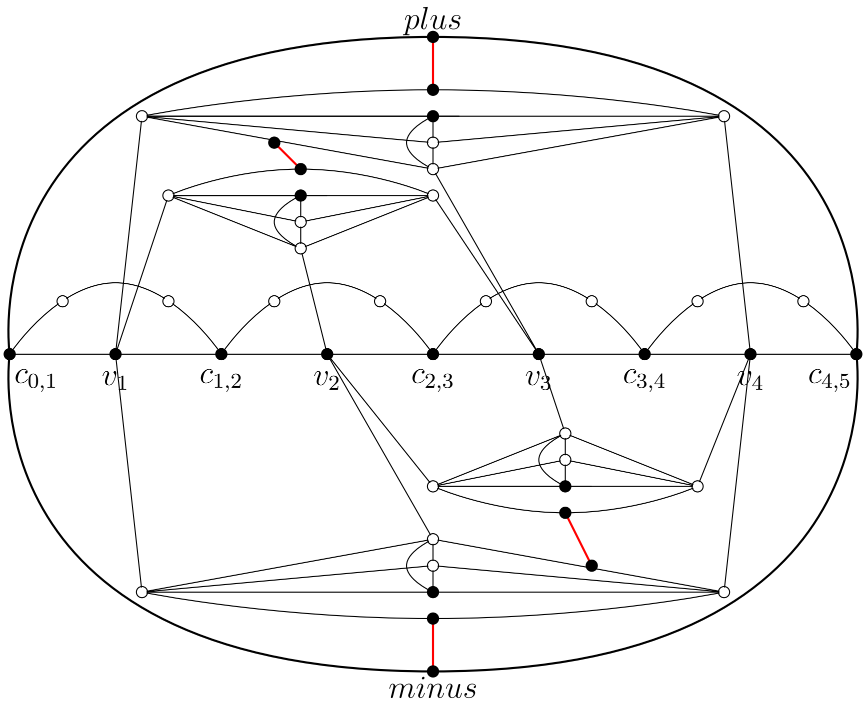



Figure 4.7: The graph $G_0$ corresponding to $X_0$. Tree structure edges are highlighted in red.

with the corresponding variable vertex and draw the variable cycle. We refer to the region bordered by the variable cycle and adjacent to the *plus* vertex as the *positive side* of the cycle, and to the region bordered by the variable cycle and adjacent to the *minus* vertex as the *negative side* of the cycle.

Draw the false literal boundaries on the positive and negative sides of the variable cycle according to the following rule. For each variable $v_i$, if $A(v_i)$ is true, draw the false literal boundary on the negative side. If $A(v_i)$ is false, draw the false literal boundary on the positive side. Figure 4.8 illustrates a drawing of the variable cycle and false literal boundaries of $G_0$ according to the assigment function $A_0 : \{v_1, v_2, v_3, v_4\} \to \{True, False\}$, which satisfies $F_0$ by assigning the variables $v_2$ and $v_3$ to $True$ and the variables $v_1$ and $v_4$ to $False$.

Each clause rectangle $C_j$ in $X$ is a connected by three edges to the variable rectangles associated with its constituent literals. Using $X$ as a template, draw each literal vertex $l_{j,k}$ at the intersection of the clause rectangle $C_j$ and the line connecting $C_j$ to the variable rectangle corresponding to $l_{j,k}$. Connect the literal vertices of $C_j$ with

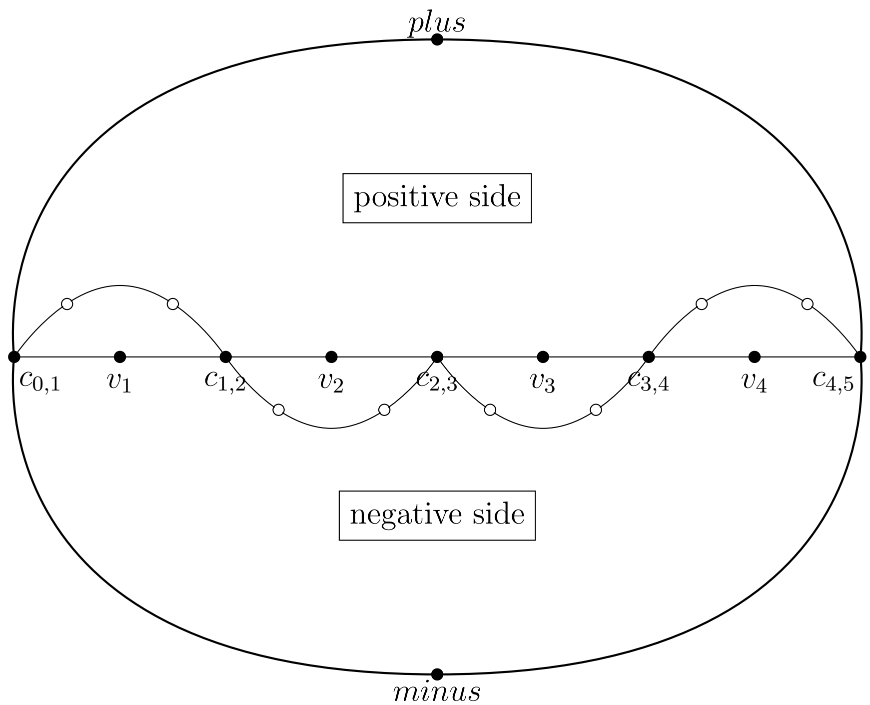



Figure 4.8: A drawing of the variable cycle of $G_0$ with false literal boundaries oriented according to $A_0$.

edges to form a triangular face. Finally, draw the vertices $closed_j$ and $open_j$ on the interior of the triangular face and connect them to the literal vertices.

Insert the tree structure edges, which by construction can be added to the drawing without creating edge crossings. Finally, connect the literal vertices to their corresponding variable vertices. Note that this creates a crossing on a false literal boundary precisely when the truth value assigned to a variable by $A$ does not match the literal. Figure 4.9 illustrates a drawing of $G_0$ according to this specification.

At this point, there exist edge crossings internal to clause gadgets and at false literal boundaries only.

### Clustering to Remove Crossings

Each crossing at a false literal boundary consists of an edge between two vertices on the boundary and an edge between a literal vertex and a variable vertex. To resolve each crossing, cluster the literal vertex with a boundary vertex as shown in Figure 4.10. For each variable $v_i$, the number of ordinary vertices on the false literal

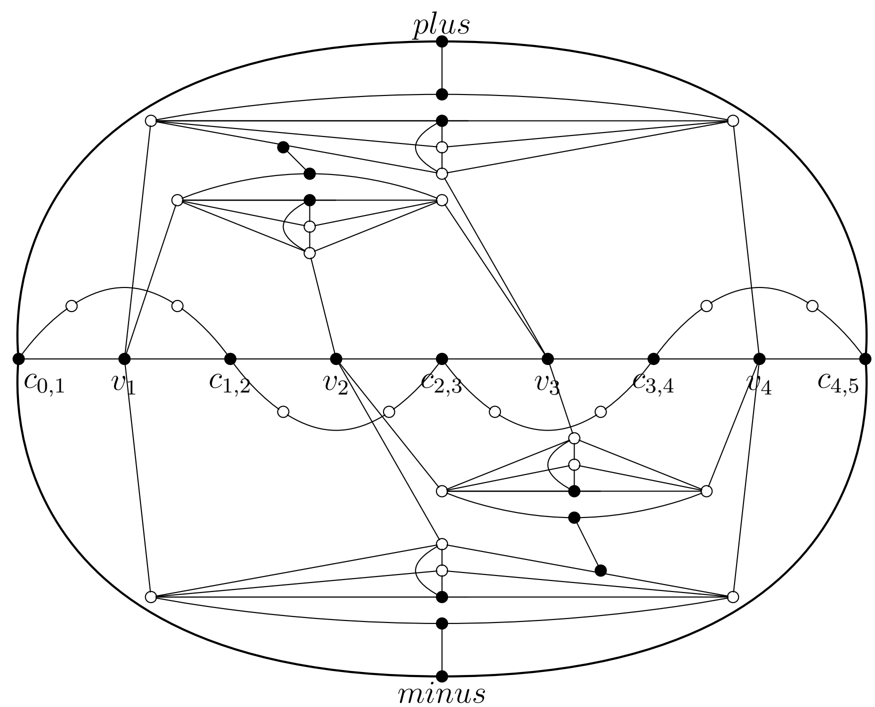



Figure 4.9: A drawing of the graph $G_0$ with crossings at false literal boundaries and inside clause gadgets.

boundary of $v_i$ is equal to $max(p_i, n_i)$, which ensures that there are enough boundary vertices to perform this operation.

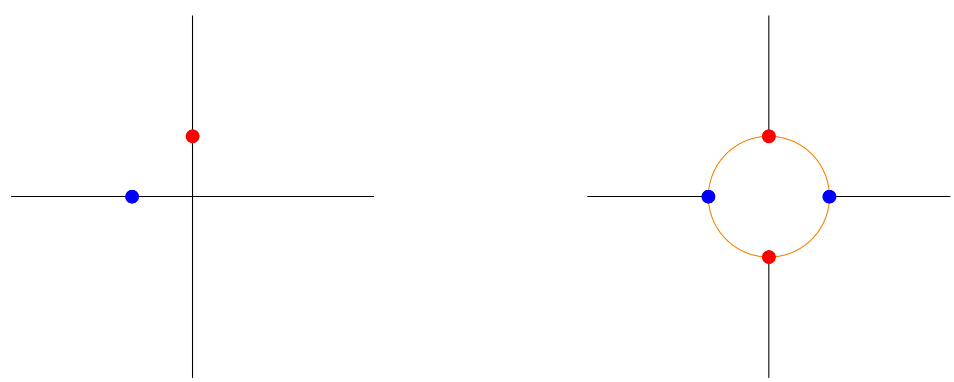


Figure 4.10: Clustering two vertices to remove a crossing at a false literal boundary.

Let $C_j$ be a positive clause. We know that $A(v_i)$ is true for at least one variable

$v_i$ corresponding to a literal $l_i$ in $C_j$ because $A$ satisfies $F$. Thus the false literal boundary of $v_i$ is drawn on the negative side of the variable cycle and the vertex $l_i$ remains unclustered. Likewise, every negative clause has at least one satisfied literal and thus every negative clause gadget has at least one unclustered literal vertex. To resolve the necessary crossing in each clause gadget, cluster an unclustered literal vertex with *open*$_j$ and draw the clause gadget according to Figure 4.6b.

The resulting drawing of $G$ is $(2,2)$-planar, and thus $G$ is a YES instance of (2,2)-Planarity. A $(2,2)$-planar drawing of $G_0$ is illustrated in Figure 4.11.

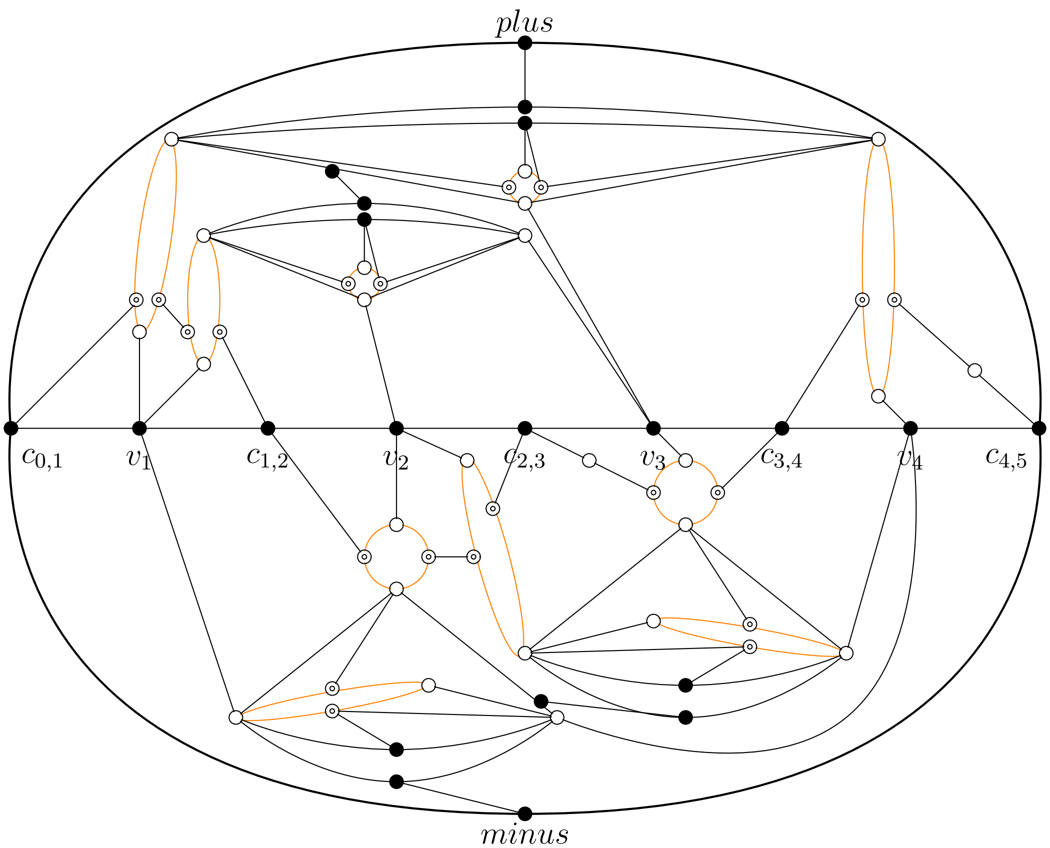



Figure 4.11: A $(2,2)$-planar drawing of the graph $G_0$.

The result of this process is a (2,2)-planar drawing of $G$, and $G$ is thus a YES instance of (2,2)-Planarity.

### 4.2.3 If $G$ Is a YES Instance of (2,2)-Planarity, $X$ is a YES Instance of Planar Monotone 3-SAT

Suppose $G$ is a YES instance of (2,2)-Planarity corresponding to an instance $X$ of Planar Monotone 3-SAT. Let $\Gamma$ be a (2,2)-planar drawing of $G$. To prove that $X$ is a YES instance, we first establish the following lemmas.

**Lemma 6.** *Let $G$ be a graph containing $C$, a cycle of K-vertices, and two vertices $v_1$ and $v_2$ connected by an edge. In any $(2,2)$-planar drawing of $G$, $v_1$ and $v_2$ are drawn on the same side of $C$.*

*Proof.* Let $\Gamma$ be a $(2,2)$-planar drawing of $G$. By Lemma 4, every K-vertex in $\Gamma$ is clustered within its $K_{8-}$ subgraph, and thus any $(2,2)$-planar drawing of $C$ is a closed loop of 2-clusters. If $v_1$ and $v_2$ are drawn on opposite sides of $C$, the edge $(v_1, v_2)$ creates a crossing because neither $v_1$ nor $v_2$ can be clustered with any vertex in $C$. □

Lemma 6 implies that the variable cycle remains intact in $\Gamma$. This enables us to prove that the positive and negative clause gadgets are separated in any $(2,2)$-planar drawing of $G$.

**Lemma 7.** *Let $\Gamma$ be a $(2,2)$-planar drawing of $G$, a graph created by our procedure according to an instance $X$ of Planar Monotone 3-SAT. Then the positive and the negative clause gadgets are drawn on opposite sides of the variable cycle in $\Gamma$.*

*Proof.* Because every positive clause gadget and every negative clause gadget needs access to the variable vertices, none can be drawn on the face adjacent to both *plus* and *minus*. Because the positive clause gadgets are connected by the tree structure, by Lemma 6, every positive gadget appears on the side of the variable cycle adjacent to *plus*. Similarly, every negative clause gadget appears on the side of the variable cycle adjacent to *minus*. □

Lemma 7 means that we may sensibly refer to the sides of the variable cycle with the positive and negative clause gadgets as the *positive side* and *negative side*, respectively. As a consequence of Lemma 6, each false literal boundary is drawn either on the positive or on the negative side of the cycle as well.

We construct an assignment function $A$ of variables from $\Gamma$ as follows. If the false literal boundary for $v_i$ is drawn on the negative side of the variable cycle in $\Gamma$, we set

$A(v_i) = True$. If the false literal boundary for $v_i$ is drawn on the positive side of the variable cycle in $\Gamma$, we set $A(v_i) = False$. The following lemma proves that $A$ is a satisfying assignment.

**Lemma 8.** *Let $\Gamma$ be a $(2, 2)$-planar drawing of $G$, a graph created by our procedure according to an instance $X$ of Planar Monotone 3-SAT. At least one literal vertex of each positive (negative) clause gadget is connected in $\Gamma$ to a variable vertex for which $A(v_i) = True$ ($A(v_i) = False$).*

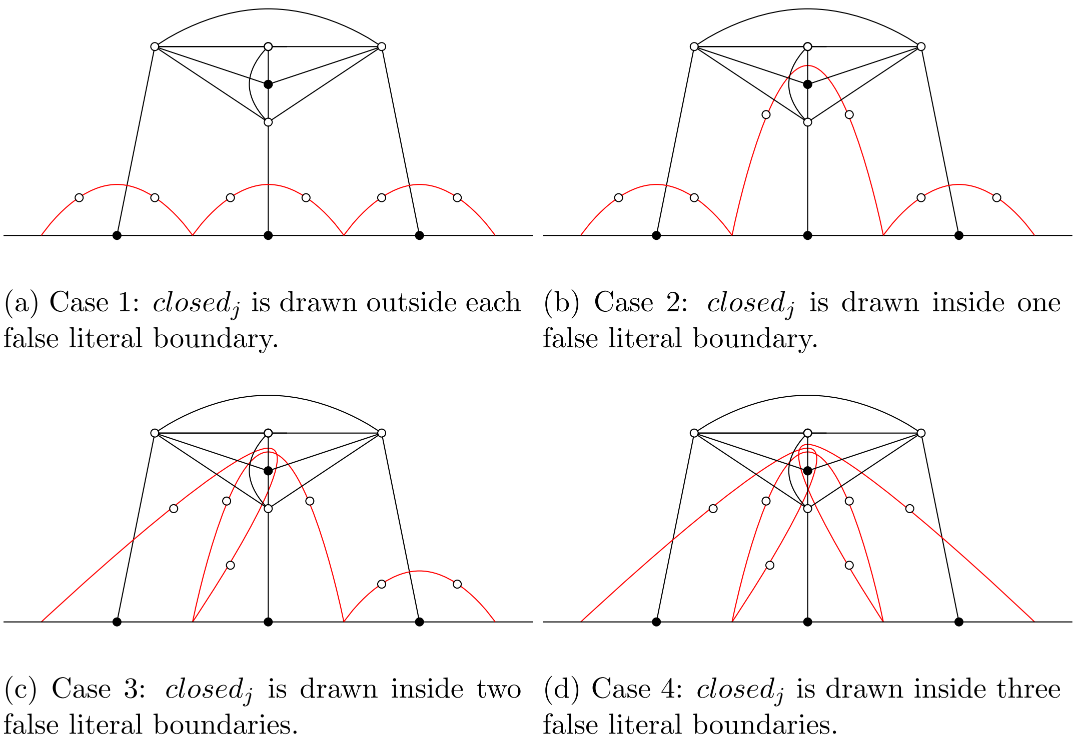
*Proof.* Without loss of generality, consider the case of a positive clause gadget $C_j$. Assume for contradiction that every literal vertex of $C_j$ is connected in $\Gamma$ to a variable $v$ with $A(v) = False$. Letting the literal vertices $l_{j1}$, $l_{j2}$, and $l_{j3}$ be connected to $v_1$, $v_2$, and $v_3$, this means that the false literal boundaries of $v_1$, $v_2$, and $v_3$ are on the positive side of the variable cycle with the clause gadget $C_j$. To prove the lemma, we prove that any placement of the vertex $closed_j$ of the clause gadget $C_j$ creates an edge crossing in $\Gamma$, a contradiction.

Because $closed_j$ is a K-vertex, it cannot be clustered with a vertex on a false literal boundary. Accordingly, we say that $closed_j$ is placed *inside* the false literal boundary of $v_i$ if it is drawn in the region between the false literal boundary and $v_i$. Otherwise, we say that $closed_j$ is placed *outside* the false literal boundary. The following cases are illustrated in Figure 4.12.

**Case 1.** Suppose $closed_j$ is placed on the positive side of the variable cycle outside the false literal boundaries of $v_1$, $v_2$, and $v_3$. Thus, for the edges ($l_{j1}$, $v_1$), ($l_{j2}$, $v_2$), and ($l_{j3}$, $v_3$) to be drawn, each literal vertex must be 2-clustered with a vertex on one of the three false literal boundaries. However, this arrangement means that the clause gadget cannot be (2,2)-planar drawn, by Lemma 5.

**Case 2.** Suppose $closed_j$ is drawn on the positive side of the variable cycle inside the false literal boundary of exactly one constituent variable. Without loss of generality, suppose $closed_j$ is drawn inside the false literal boundary of $v_1$. In this case, the path ($closed_j$, $l_{j2}$, $v_2$) intersects the false literal boundaries associated with $v_1$ and $v_2$. Because $closed_j$ and $v_2$ are K-vertices, only $l_{j2}$ can be clustered with a false literal boundary vertex and thus this placement creates at least one necessary crossing.

**Case 3.** Suppose $closed_j$ is drawn on the positive side of the variable cycle inside the false literal boundary of exactly two constituent variables. Without loss of



(a) Case 1: $closed_j$ is drawn outside each false literal boundary.

(b) Case 2: $closed_j$ is drawn inside one false literal boundary.

(c) Case 3: $closed_j$ is drawn inside two false literal boundaries.

(d) Case 4: $closed_j$ is drawn inside three false literal boundaries.

Figure 4.12: Possible placements of the clause vertex $closed_j$ relative to three clause boundaries.

generality, suppose $closed_j$ is drawn inside the false literal boundaries of $v_1$ and $v_2$. $closed_j$ is drawn outside the false literal boundary of $v_3$, so the path $(closed_j, l_{j3}, v_3)$ crosses the false literal boundaries of $v_1$, $v_2$, and $v_3$. Because the only variable in the path that can be 2-clustered is $l_{j3}$, there is at least one necessary crossing.

**Case 4.** Suppose $closed_j$ is drawn on the positive side of the variable cycle inside the false literal boundary of all three of its constituent variables. In this case, the path $(closed_j, l_{j1}, v_1)$ intersects the false literal boundaries of $v_2$ and $v_3$. Because the only variable in the path that can be 2-clustered is $l_{j1}$, there is at least one necessary crossing.

Thus if every literal vertex of $C_j$ is connected in $\Gamma$ to a variable $v$ with $A(v) = False$, a necessary crossing is created in $\Gamma$ regardless of the position of the a vertex $closed_j$. This creates a contradiction, which suffices to show that at least one of the literal vertices of $C_j$ must match the assignment of its associated variable vertex. $\square$

By Lemma 8, at least one literal vertex $l_i$ of each clause gadget $C_j$ in $\Gamma$ is connected to a variable $v_i$ with $A(v_i) = l_i$. Thus $A$ is a satisfying assignment for $F$, and $X$ is a YES instance of Planar Monotone 3-SAT.

Together, sections 4.2.1 and 4.2.2 establish Theorem 15.

## 4.3 Conclusion

In this chapter, we proved that the question of $(k, 1)$-planarity can be easily decided when $k \leq 3$ or a clustering is fixed. However, we also demonstrated that deciding $(4, 1)$-planarity and $(2, 2)$-planarity is NP-complete. For larger values of $k$ and $p$, the hardness of deciding $(k, p)$-planarity remains unknown. We conjecture that the $(k, p)$-planarity decision problem is NP-complete in these cases.

# Chapter 5

# Intracluster Representations

As demonstrated in Chapters 2 and 3, $(k, p)$-planar drawings allow large classes of nonplanar graphs to be represented in the plane without edge crossings. However, the choice to represent a graph with a $(k, p)$-planar drawing has necessary costs. First, a $(k, p)$-planar drawing allows the edges incident to each particular vertex to be divided among up to $p$ ports, making it less apparent that the incident edges are adjacent to each other. Second, a $(k, p)$-planar drawing omits intracluster edges entirely. We can address these problems either by constructing our $(k, p)$-planar graphs in accordance with external stipulations (such as the requirement that all clusters contain complete subgraphs) or by adding additional notation to the $(k, p)$-planar drawing. In this chapter, we pursue the latter approach.

This chapter considers strategies of *intracluster representation*, which address the problems of port-divided incident edges and intracluster structure by creating representations inside each cluster region. For example, an intracluster representation might identify ports of the same vertex by drawing arcs between them or represent intracluster structure with an intersection representation. In general, intracluster representations strike a balance between simplicity and accuracy: by including more information inside each cluster region, we elucidate the structure of the graph but make our drawing more visually complex.

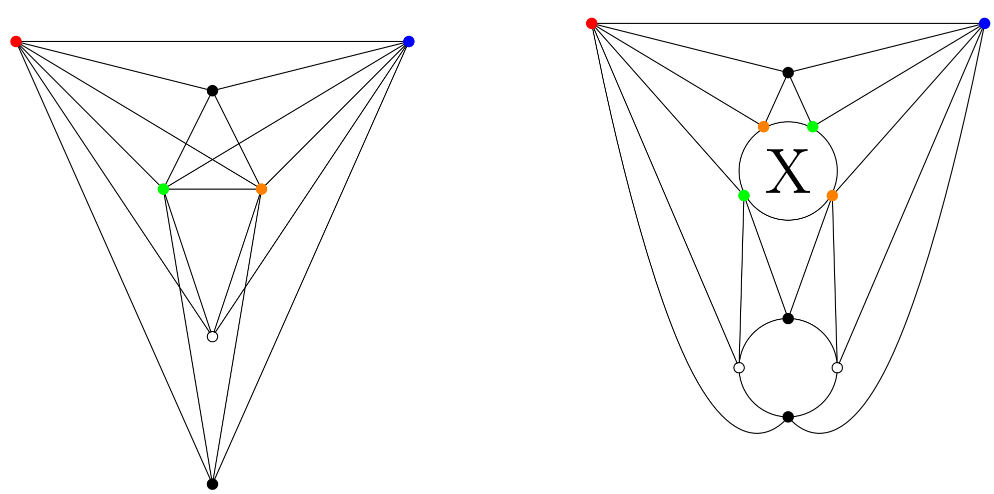
Section 5.1 considers $(2, p)$-planar drawings with marked crossings, a simple intracluster representation scheme that allows the $(2, p)$-planar graphs to be drawn on the plane without losing any information. We then consider $(k, 2)$-planar drawings with intracluster circle representations, $(k, p)$-planar drawings with intracluster polygon-circle representations, and $(k, 4)$-planar drawings with intracluster adjacency matrices (also called $k$-NodeTrix drawings) in Sections 5.2-5.4. Finally, we consider flexible and permissive representations in Section 5.5.

## 5.1 $(2, p)$-Planar Drawings with Marked Crossings

In a $(2, p)$-planar drawing, each cluster contains at most two vertices which may or may not be connected by an intracluster edge. We can communicate the presence or absence of this edge with a binary indicator regardless of the number of ports on the perimeter of the cluster region.

A $(2, p)$-planar drawing with marked crossings is a $(2, p)$-planar drawing in which every cluster of two adjacent vertices is labeled with an 'X'. (Alternatively, in a $(2, 2)$-planar drawing, the adjacency of two clustered vertices can be represented by drawing an internal edge between each pair of ports.) Clusters containing non-adjacent vertices are left unlabeled.

Figure 5.1 illustrates a $(2, 2)$-planar graph and a corresponding $(2, 2)$-planar drawing with marked crossings.




(a) A nonplanar, $(2, 2)$-planar graph $G$.

(b) A $(2, 2)$-planar drawing of $G$ with marked crossings.

Figure 5.1: A graph $G$ and corresponding $(2, 2)$-planar drawing with marked crossings.

As marked crossings can be added to every $(2, p)$-planar drawing, they provide a convenient way to draw every $(2, p)$-planar graph on the plane without losing information. However, although the method of marking crossings is convenient, it does not scale to clusters of more than two vertices. The following section considers a method of intracluster representation that is compatible with $(k, 2)$-planar drawings for all positive integers $k$.

## 5.2 $(k, 2)$-Planar Drawings with Intracluster Circle Representations

As the edge density of a graph increases, traditional node-link representations become more visually complex. Intersection representations, which represent vertices as geometric objects and indicate an edge whenever two objects overlap, provide an alternative in this case.

In a circle graph representation, each vertex is represented by a chord on a circle and each edge is represented by an intersection of chords. Circle graphs can be recognized in $O(n^2)$ time [27]. Although not every graph is a circle graph, the circle graphs include many nonplanar graphs, including every complete graph.

We define a $(k, 2)$-circle-planar drawing as a $(k, 2)$-planar drawing in which the two ports of each clustered vertex are connected by a chord within a cluster region and each cluster region is a circle graph that accurately represents its intracluster structure. Figure 5.2 illustrates a $(4, 2)$-circle-planar drawing of $K_6$.

The stipulation that each cluster region is a circle graph ensures that a $(k, 2)$-circle-planar drawing contains all the information required to recreate the original graph. However, although all $(k, 2)$-circle-planar graphs are trivially $(k, 2)$-planar, not all $(k, 2)$-planar graphs are $(k, 2)$-circle-planar. We will prove the specific case that there exists a $(2, 2)$-planar graph that is not $(2, 2)$-circle-planar.

**Proposition 4.** *There exists a* $(2, 2)$*-planar graph that is not* $(2, 2)$*-circle planar.*

*Proof.* Figure 5.3a illustrates a graph $G$ with a $K_{3,3}$ minor. In the figure, white vertices represent ordinary vertices and black vertices abbreviate K-vertices and associated $K_{8-}$ subgraphs as described in Section 4.2. $G$ admits the $(2, 2)$-planar drawing illustrated in Figure 5.3b.

Suppose for contradiction that $G$ admits a $(2, 2)$-circle-planar drawing $\Gamma$. Lemma 4, proved in Section 4.2, states that any graph that contains a $K_{8-}$ subgraph admits

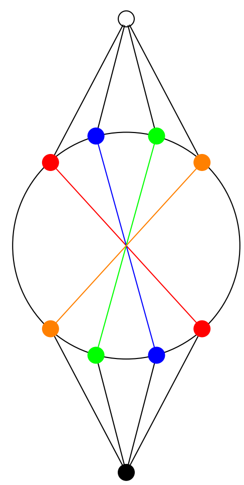
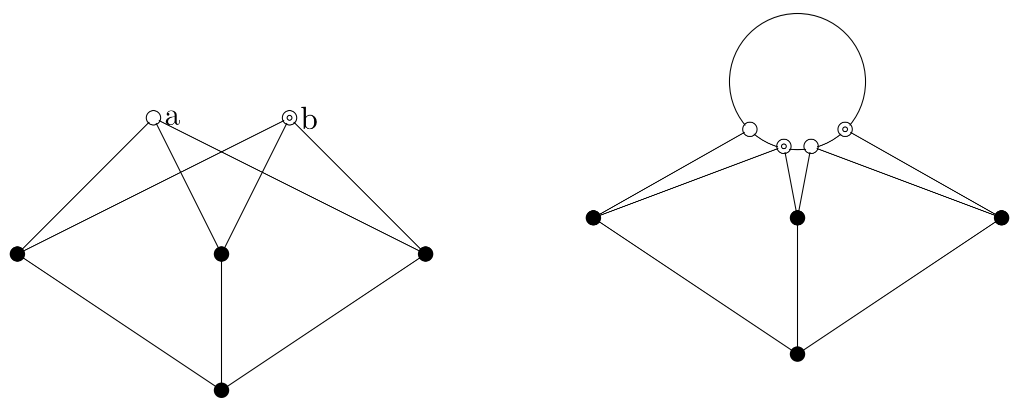


Figure 5.2: A $(4,2)$-circle-planar drawing of the complete graph $K_6$.




(a) A $(2,2)$-planar graph $G$.

(b) A $(2,2)$-planar drawing of $G$. Black vertices abbreviate the $(2,2)$-planar drawing of $K_{8-}$ illustrated in Figure 4.3.

Figure 5.3: A $(2,2)$-planar graph $G$ that is not $(2,2)$-circle planar. K-vertices and their associated $K_{8-}$ subgraphs are drawn as solid black dots.

only $(2,2)$-planar drawings in which the K-vertex $v$ is clustered within its $K_{8-}$ subgraph. The proof entails that every vertex in the $K_{8-}$ subgraph must be clustered with another vertex in the subgraph.

Suppose the vertices $a$ and $b$ are left unclustered in $\Gamma$. In this case, the contracted graph $G_C$ of $G$ contains a $K_{3,3}$ minor and is thus nonplanar. This creates a contradiction, as $G_C$ is planar if $G$ is $(2, 2)$-planar.

Alternatively, suppose that $a$ and $b$ are included in the same cluster in $\Gamma$. In this case, we may assume without loss of generality that each is represented by two ports on the boundary of a cluster region $R$. Because $a$ and $b$ are non-adjacent, their ports in $\Gamma$ must not alternate to ensure the accuracy of the intracluster circle graph. Thus the two ports of $a$ and the two ports of $b$ are adjacent along the perimeter of $R$ in $\Gamma$.

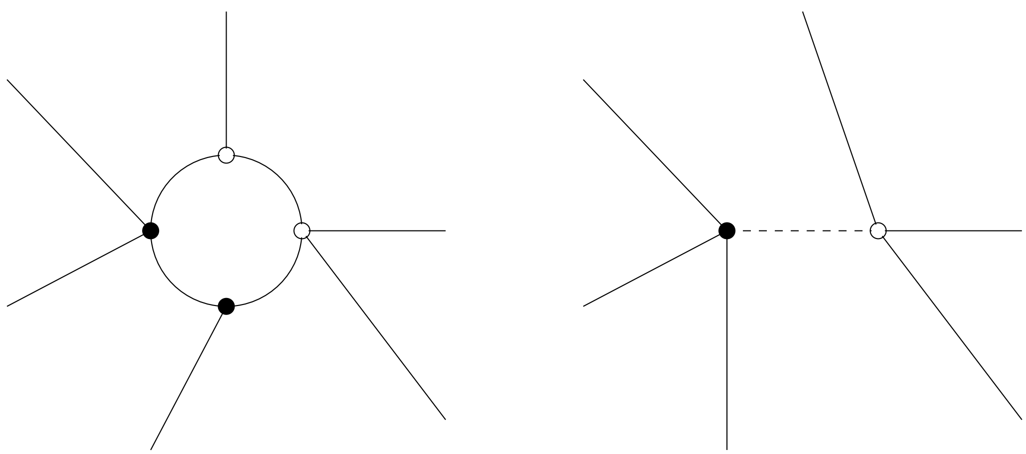


Figure 5.4: Eliminating a 2-cluster with adjacent ports.

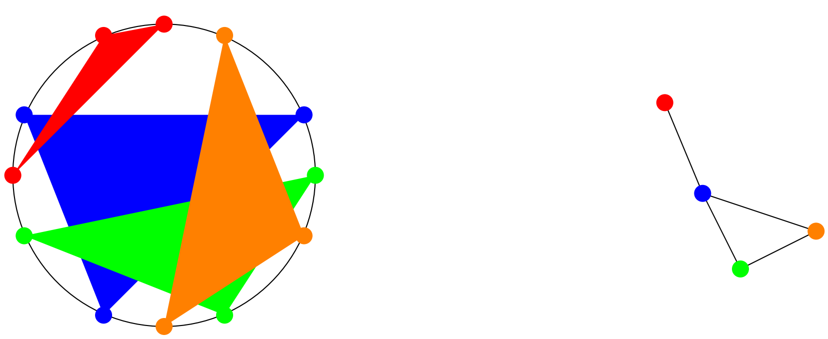
However, a 2-cluster with adjacent ports is superfluous. To see this, note that in a 2-cluster with adjacent ports, the edges incident to each vertex can be consolidated and the cluster region subsequently removed as illustrated in Figure 5.4. By this process, $\Gamma$ can be transformed into a a $(2, 2)$-planar drawing of $G$ in which $a$ and $b$ are unclustered. This is a contradiction, and thus no $(2, 2)$-planar-circle drawing of $G$ is possible. $\square$

We note that $G$ could instead be represented by a $(2, 2)$-planar drawing with marked crossings. In general, because the requirements of $(k, 2)$-circle-planarity limit the ways in which ports can be placed, the classes of $(k, 2)$-circle-planar graphs are smaller than the classes of $(k, 2)$-planar graphs. Determining the precise relationship between the two remains an open problem.

## 5.3 $(k,p)$-Planar Drawings with Intracluster Polygon-Circle Representations

A polygon-circle graph is the intersection graph of a set of polygons inscribed in a circle, and thus may be considered a logical extension of the circle graph. For instance, a triangle-circle graph is the intersection graph of a set of triangles inscribed in a circle. Recognition of polygon-circle graphs is NP-complete [25].

Every cluster with $p$ ports per vertex thus corresponds to a $p$-gon-circle graph in the same way that every cluster with two ports per vertex corresponds to a circle graph. For example, Figure 5.5 illustrates a 4-cluster with inscribed triangles and the corresponding triangle-circle graph. Alternatively, we may view Figures 5.5a and 5.5b as triangle-circle and node-link representations of the same graph.



(a) A cluster region $R_i$ with inscribed triangles.

(b) The triangle-circle graph of $R_i$.

Figure 5.5: A 4-cluster and corresponding triangle-circle graph.

Polygon-circle graphs have key advantages over circle graphs. First, the class of $(n+1)$-gon-circle graphs generalizes the class of $n$-gon-circle graphs. This is apparent from the observation that an $n$-gon-circle graph can be transformed into a $(n+1)$-gon-circle graph by adding a trivially small edge to each inscribed $n$-gon. Every circle graph is thus a triangle-circle graph.

Moreover, there exist polygon-circle graphs that are not circle graphs. In [6], Andre Bouchet notes that the wheel graph $W_6$ with five spokes is not a circle graph. However, as illustrated in Figure 5.6, $W_6$ is in fact a triangle-circle graph. Furthermore, intracluster polygon-circle graphs are not limited to $(k, p)$-planar graphs with 2 ports per vertex, as inscribed polygons may have an arbitrary number of vertices.

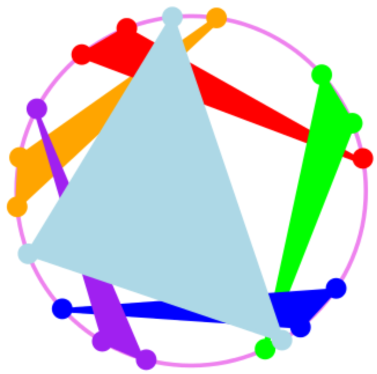


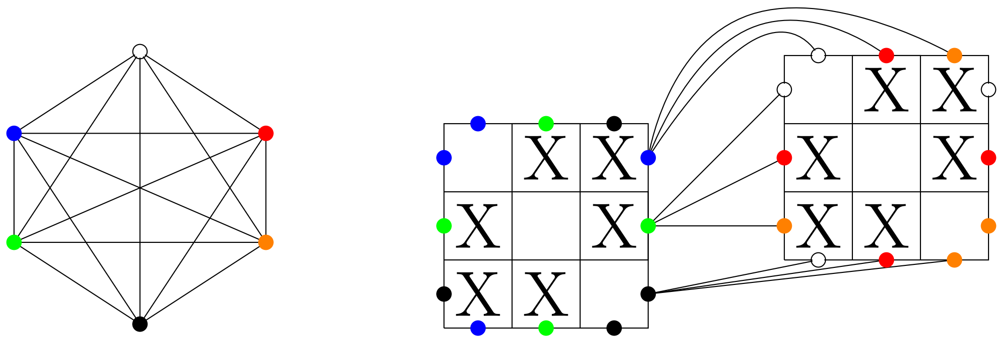
Figure 5.6: A triangle-circle representation of the wheel graph $W_6$.

However, polygon-circle graphs have several of the same limitations as circle graphs. Depending on how the requirements for polygon intersection constrain port ordering on the outside of cluster regions, $(k, p)$-planar graphs may not be $(k, p)$-polygon-circle-planar. The problem of determining which $(k, p)$-planar graphs are $(k, p)$-polygon-circle-planar remains an open problem. In the next section, we consider an intracluster representation compatible with any cluster subgraph.

## 5.4 $(k, p)$-Planar Drawings with Intracluster Adjacency Matrices

In addition to inscribed polygons, adjacency matrices can be drawn inside cluster regions to completely represent the intracluster structure. In a $(k, p)$-adjacency-matrix-planar drawing, the inside of each $k$-cluster contains a $k \times k$ adjacency matrix that displays the cluster's internal structure. Ports are placed at both ends of the row and column of the adjacency matrix corresponding to each vertex. A $(k, p)$-planar drawing with intracluster adjacency matrices thus requires four ports per cluster (except when $k = 2$, in which case the number of ports per cluster is effectively three.) Figure 5.7 depicts $K_6$, which is $(3, 4)$-adjacency-matrix planar, and a $(3, 4)$-adjacency-matrix planar drawing of $K_6$.

The idea of using adjacency matrices to represent clusters was advanced by Henry,



(a) $K_6$. (b) A 3-NodeTrix-planar drawing of $K_6$.

Figure 5.7: Node-link and NodeTrix-planar representations of $K_6$.

Fekete, and McGuffin in [18]. The authors referred to their system as *NodeTrix*, and thus when Di Giacomo, Liotta, Patrigniani and Tappini [13] formalized the $(k, p)$-adjacency-matrix-planar drawing, they used the term *$k$-NodeTrix planar* to refer to a $(k, p)$-adjacency-matrix planar graph. For the remainder of this section, we will use their nomenclature for convenience.

Every intracluster graph can be represented by an adjacency matrix, whereas intersection representations such as polygon-circle representations apply to limited classes of graphs. However, adjacency matrices convey less of the structure of the graph as an immediate impression. Moreover, $k$-NodeTrix planarity requires that cluster regions are limited to four ports per vertex.

The class of $k$-NodeTrix-planar graphs is still somewhat restrictive because the ports corresponding to a particular vertex must be placed at the ends of the corresponding row and column of the adjacency matrix. Although the rows and columns can be shuffled, this restriction on port orderings means that not all $(k, 4)$-planar graphs are $k$-NodeTrix planar. Di Giacomo et al. [13] show that the problem of testing $k$-NodeTrix-planarity for cluster graphs can be solved in $O(n^3)$ time when $k = 2$ and is NP-complete for $k \geq 3$. Because this result relies on the limitations on port ordering imposed by the intracluster adjacency matrix, it does not directly imply the NP-completeness of testing $(k, 4)$-planarity for cluster graphs when $k \geq 3$.

If the requirements of $k$-NodeTrix planarity are still too stringent for a particular application, even more flexible intracluster representations are possible. We consider such representations in our final section.

## 5.5 Permissive Intracluster Representations

We refer to any intracluster representation that does not rely on the order or number of ports on the border of the cluster region as a *permissive intracluster representation*. For example, the marked crossings considered in Section 5.1 are permissive.

Discarding the restriction that an intracluster representation must relate to the exterior surface of its cluster region allows a huge variety of possibilities. For instance, the structure of a cluster subgraph could be drawn using an ordinary node-link representation, which would be particularly effective if the subgraph in question were planar but not outerplanar. Alternatively, cluster subgraphs could be given intracluster $(k, p)$-planar drawings, opening the door to recursive representation schemas. The intracluster representations previously discussed, including circle-polygon graphs and adjacency matrices, can be applied permissively. In particular, because adjacency matrices can be used to represent any graph, permissive intracluster adjacency matrices can be employed to capture the structure of any $(k, p)$-planar graph in full.

The obvious drawback of permissive intracluster representations is loss of readability. In a permissive intracluster representation, ports on cluster boundaries are not associated with their intracluster vertex representations except perhaps by common colors or labels, and thus the connections between intercluster and intracluster elements may be obscured. For this reason, permissive intracluster representations can be used most effectively when each cluster represents a distinct semantic unit. If the fact that two clusters are connected is more important than which two vertices manifest the connection, a permissive intracluster representation may be wholly adequate.

Finally, permissive schemes for representing intracluster structure might mix and match intracluster representations as necessary in order to be most visually effective. Such elaborate schemes are probably best discussed on an individual basis using the vocabulary of graphic design.

## 5.6 Conclusion

In this chapter, we discussed a variety of methods for reintroducing the intracluster structure elided in a $(k, p)$-planar drawing. These methods ranged from the most readable but most narrowly applicable, such as $(2, p)$-planar drawings with marked crossings, to the most broadly applicable but difficult to quickly interpret, such as permissive intracluster representations with mixed intracluster representations.

More work remains to be done before the potential of intracluster representations is fully understood. However, much of this work has to do with the ease of visualization and is more practical than theoretical. We hope that the overview provided in this chapter is sufficient to convince the reader of the practical applicability of $(k, p)$-planar drawings combined with intracluster representations.

# Chapter 6

# Summary and Future Directions

In this chapter, we review the results proved in previous chapters and comment on their significance to the broader goals of this work. Additionally, we state open problems and outline promising directions for future research.

## 6.1 Review of Results

We began this thesis by providing several answers to the question "Why do we need a new representation for cluster graphs?" First, we observed the difficulty of representing small-world networks, graphs with mined substructures, and external partition graphs with traditional node-link representations. We reviewed the related literature, including several graph representations that addressed aspects of our problem with varying degrees of success. Finally, we argued that $(k, p)$-planar drawings successfully meet the demands of each of our use cases and generalize existing cluster graph representations. A formal understanding of the $(k, p)$-planar graphs thus improves our ability to represent several graph types and provides insight into the existing representations generalized by $(k, p)$-planar drawings.

We focused on this formal understanding for the remainder of the work, pursuing several lines of investigation. First, we sought to understand the $(k, p)$-planar graphs in relation to established graph classes. We proved that there exist small values of $k$ and $p$ for which the $(k, p)$-planar graphs are equivalent to the planar graphs and to the IC-Planar graphs. Our inquiry suggested a promising conclusion: as we increase $k$ and $p$, the class of $(k, p)$-planar graphs grows rapidly.

How rapidly does the class of $(k, p)$-planar graphs grow as we increase $k$ and $p$?

One way to measure this quantity is by considering the maximum number of edges in a $(k, p)$-planar graph. Euler's edge bound indicates that for each vertex we add to a planar graph, the maximum number of edges increases by 3. We proved Theorems 8 and 10, which tell us that when $k$ is fixed, each additional cluster increases the maximum number of edges by $4k^2 - k$ or $kp + k^2 - k$, depending on the value of $p$. We also proved Theorem 13, which tightly bounds the number of edges in a $(k, p)$-planar graph based on its number of vertices, and Theorem 12, which requires a specified clustering but provides an even more precise bound on the number of edges in a $(k, p)$-planar graph.

Next, we turned to the hardness of deciding whether or not a graph is $(k, p)$-planar. Although it is possible to determine whether a cluster graph $G$ is $(k, 1)$-planar in linear time, the problem appears much harder for larger values of $k$ and $p$. When the clustering is left unspecified, we proved that it is NP-complete to decide whether or not a graph $G$ is $(4, 1)$-planar or $(2, 2)$-planar, and speculated that the $(k, p)$-planar decision problem remains NP-complete for larger values of $k$ and $p$.

Finally, we addressed the practical matter of tailoring $(k, p)$-planar graphs to convey more information effectively with intracluster representations. We explored a range of options, from intracluster representations that optimized for simplicity and readability to those that sacrificed design coherence for flexibility. At one end of the spectrum, we demonstrated examples such as intracluster polygon-circle representations and adjacency matrices, which used the same ports to represent vertices on the interior and exterior of the cluster region. At the other extreme, we noted the freedom inherent in leaving the interior of the cluster region unspecified. If necessary, we can inscribe a different representation in each cluster region to create a hybrid drawing tailored for a specific application.

## 6.2 Future Directions

In previous chapters, we noted several open problems that represent promising avenues for future work. First, the $(k, p)$-planar graphs can still be better related to existing graph classes. We suspect that there exist integers $k$ and $p$ such that every NIC-planar graph is $(k, p)$-planar. More broadly, we observed a divergence between the classes of $(k, p)$-planar graphs and several established graph classes as $k$ and $p$ increased. Do there exist additional values $k$ and $p$ for which the $(k, p)$-planar graphs are contained within or equivalent to an established class of graphs?

We have observed that Theorem 13, which bounds the number of edges in a $(k, p)$-planar graph according to the number of vertices, is tight in the $p = 1$ case. Moreover, the proof of Theorem 13 parallels the proof of Theorem 10 on a more detailed scale. Accordingly, we conjecture that Theorem 13 is tight in the case where $k > p > 0$. Establishing this conjecture in the affirmative would make the relationship between the number of vertices and the maximum number of edges in a $(k, p)$-planar graph precise in the $k > p$ case.

In Chapter 4, we prove that deciding $(4, 1)$-planarity and deciding $(2, 2)$-planarity are NP-complete problems. For larger values of $k$ and $p$, the $(k, p)$-planarity decision problem appears even more complex. If we are correct in our supposition, then the $(k, p)$-planarity decision problem is NP-complete for large $k$ and $p$. However, it remains possible that for certain values of $k$ and $p$, the class of $(k, p)$-planar graphs is easily decidable. For fixed values of $k$ and $p$, how hard is the $(k, p)$-planarity decision problem?

Finally, each of the non-permissive intracluster representations presented in Chapter 5 presents a series of natural research questions. For instance, how large is the class of $k$-NodeTrix planar graphs? Is it possible to decide in polynomial time whether a graph is $(k, 2)$-circle-planar? Additionally, the intracluster representations presented in the chapter are far from the only possibilities. Further research might define new intracluster representations or compare the merits of intracluster representations in a practical context.

# Bibliography


[1] Noga Alon, Raphael Yuster, and Uri Zwick. “Color-coding”. In: *Journal of the ACM* 42.4 (1995), pp. 844–856.

[2] Noga Alon, Raphael Yuster, and Uri Zwick. “Finding and counting given length cycles”. In: *Algorithmica* 17.3 (1997), pp. 209–223.

[3] Patrizio Angelini, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Maurizio Patrignani, and Ignaz Rutter. “Intersection-link representations of graphs”. In: *International Symposium on Graph Drawing and Network Visualization*. Springer. 2015, pp. 217–230.

[4] Vladimir Batagelj, Franz J Brandenburg, Walter Didimo, Giuseppe Liotta, Pietro Palladino, and Maurizio Patrignani. “Visual analysis of large graphs using (X, Y)-clustering and hybrid visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.11 (2011), pp. 1587–1598.

[5] Mark de Berg and Amirali Khosravi. “Optimal binary space partitions in the plane”. In: *International Computing and Combinatorics Conference*. Springer. 2010, pp. 216–225.

[6] André Bouchet. “Circle graph obstructions”. In: *Journal of Combinatorial Theory*. B 60.1 (1994), pp. 107–144.

[7] Franz J Brandenburg, Walter Didimo, William S Evans, Philipp Kindermann, Giuseppe Liotta, and Fabrizio Montecchiani. “Recognizing and drawing IC-planar graphs”. In: *Theoretical Computer Science* 636 (2016), pp. 1–16.

[8] Diane J Cook and Lawrence B Holder. “Substructure discovery using minimum description length and background knowledge”. In: *Journal of Artificial Intelligence Research* 1 (1994), pp. 231–255.

[9] William Cook. *The Traveling Salesman Problem*. Dec. 2016. URL: `http://www.math.uwaterloo.ca/tsp/problem/index.html`.

[10] Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, and Maurizio Patrignani. "Computing NodeTrix representations of clustered graphs". In: *International Symposium on Graph Drawing and Network Visualization*. Springer. 2016, pp. 107–120.

[11] Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Pietro Palladino. "Visual analysis of one-to-many matched graphs". In: *International Symposium on Graph Drawing*. Springer. 2008, pp. 133–144.

[12] Emilio Di Giacomo, William J Lenhart, Giuseppe Liotta, Timothy W Randolph, and Alessandra Tappini. *(k,p)-Planar Graphs: A Generalization of Hybrid Planarity Models*. In Preparation.

[13] Emilio Di Giacomo, Giuseppe Liotta, Maurizio Patrignani, and Alessandra Tappini. "NodeTrix planarity testing with small clusters". In: *arXiv preprint arXiv:1708.09281* (2017).

[14] David Eppstein, Philipp Kindermann, Stephen Kobourov, Giuseppe Liotta, Anna Lubiw, Aude Maignan, Debajyoti Mondal, Hamideh Vosoughpour, Sue Whitesides, and Stephen Wismath. "On the planar split thickness of graphs". In: *Algorithmica* 80.3 (2018), pp. 977–994.

[15] Leonhard Euler. "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii Academiae Scientiarum Petropolitanae* 8 (1741), pp. 128–140.

[16] Huahai He and Ambuj K Singh. "Efficient algorithms for mining significant substructures in graphs with quality guarantees". In: *IEEE International Conference on Data Mining*. IEEE. 2007, pp. 163–172.

[17] Nathalie Henry, Anastasia Bezerianos, and Jean-Daniel Fekete. "Improving the readability of clustered social networks using node duplication". In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1317–1324.

[18] Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. "NodeTrix: A hybrid visualization of social networks". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1302–1309.

[19] John Hopcroft and Robert Tarjan. "Efficient planarity testing". In: *Journal of the ACM* 21.4 (1974), pp. 549–568.

[20] Petra Isenberg, Sheelegh Carpendale, Anastasia Bezerianos, Nathalie Henry, and Jean-Daniel Fekete. "CoCoNutTrix: Collaborative retrofitting for information visualization". In: *IEEE Computer Graphics and Applications* 29.5 (2009), pp. 44–57.

[21] Donald E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions (Art of Computer Programming)*. 1st ed. Addison-Wesley Professional, 2008. ISBN: 0321534964, 9780321534965.

[22] Donald E Knuth and Arvind Raghunathan. "The problem of compatible representatives". In: *SIAM Journal on Discrete Mathematics* 5.3 (1992), pp. 422–427.

[23] Vladimir P Korzhik. "Minimal non-1-planar graphs". In: *Discrete Mathematics* 308.7 (2008), pp. 1319–1327.

[24] Mark E Newman. "The structure and function of complex networks". In: *SIAM Review* 45.2 (2003), pp. 167–256.

[25] Martin Pergel. "Recognition of polygon-circle graphs and graphs of interval filaments is NP-complete". In: *International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer. 2007, pp. 238–247.

[26] Satu Elisa Schaeffer. "Graph clustering". In: *Computer Science Review* 1.1 (2007), pp. 27–64.

[27] Jeremy Spinrad. "Recognition of circle graphs". In: *Journal of Algorithms* 16.2 (1994), pp. 264–282.

[28] Duncan J Watts and Steven H Strogatz. "Collective dynamics of "small-world" networks". In: *Nature* 393.6684 (1998), pp. 440–442.

[29] Xifeng Yan and Jiawei Han. "gspan: Graph-based substructure pattern mining". In: *IEEE International Conference on Data Mining*. IEEE. 2002, pp. 721–724.

[30] Xin Zhang. "Drawing complete multipartite graphs on the plane with restrictions on crossings". In: *Acta Mathematica Sinica, English Series* 30.12 (2014), pp. 2045–2053.

[31] Xin Zhang, Guizhen Liu, and Yong Yu. "On (p, 1)-total labelling of plane graphs with independent crossings". In: *Filomat* 26.6 (2012), pp. 1091–1100.