



Roma Tre University
Ph.D. in Computer Science and Automation

Graph Analysis via Induced Subgraphs

Marco D'Elia

Graph Analysis via Induced Subgraphs

A thesis presented by

Marco D'Elia

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Computer Science and Automation

Roma Tre University

Dept. of Civil, Computer Science and Aeronautical Technologies Engineering

Spring 2026

ADVISOR:

Prof. Maurizio Patrignani, Roma Tre University, Italy

CO-ADVISORS:

Prof. Irene Finocchi, LUISS Guido Carli, Italy

Prof. Fabrizio Frati, Roma Tre University, Italy

REVIEWERS:

Prof. Andrea Marino, University of Florence, Italy

Prof. Michael J. Pelsmayer, Illinois Institute of Technology, USA

To the brightest star in the firmament.

Abstract

Induced subgraphs constitute a central tool for understanding the structure of complex networks, as they retain original adjacency relations while highlighting relevant local and global properties. This thesis investigates induced subgraphs from complementary algorithmic and combinatorial viewpoints, aiming to characterize both their practical effectiveness in network analysis and their structural limitations in sparse graph classes.

The first part of the thesis studies maximal cliques as fundamental dense induced subgraphs for graph analysis. Despite their worst-case intractability, maximal cliques can be efficiently enumerated in many sparse real-world graphs. Leveraging this property, we introduce clique-based representations for graph partitioning and community detection, in which vertices are embedded into a metric space according to their clique participation, enabling the use of standard clustering algorithms. This approach produces high-quality partitions and compares favorably with established community detection methods. To address the inherent redundancy and abundance of maximal cliques, we further introduce a unified framework for clique summarization. Within this framework, we formalize several principled summarization criteria and problem variants, and develop efficient algorithms for identifying isolated cliques, supported by pruning techniques that substantially reduce the enumeration search space.

The second part of the thesis focuses on extremal questions concerning induced subgraphs in planar graphs. We derive bounds on the size of induced subgraphs with bounded maximum degree in planar and outerplanar graphs, obtaining tight results for specific parameter values. Additionally, we investigate the guaranteed existence of large outerplanar induced subgraphs in 2-outerplanar graphs, for which we establish a tight bound, and in planar 3-trees, where we derive both lower and upper bounds that extend and refine classical results in planar graph theory.

Together, these contributions provide a coherent and unified perspective on induced subgraphs as a versatile tool for algorithmic network analysis and for the study of structural properties in sparse graphs.

Contents

Introduction	1
1 Preliminaries	7
I Maximal Cliques as Induced Subgraphs	13
2 High-Modularity Graph Partitioning Through NLP Techniques and Maximal Clique Enumeration	15
2.1 Introduction	15
2.2 Preliminaries	18
2.3 Overview of the Approach	20
2.4 Experimental Setup	24
2.5 Evaluation of CLIQUE-TF-IDF for Computing k -way Partitions	28
2.6 Evaluation of CLIQUE-TF-IDF with Unknown Number of Blocks	33
2.7 Experimental Validation of the Clique-based Approach	39
3 Maximal Cliques Summarization: Principles, Problem Classification, and Algorithmic Approaches	45
3.1 Introduction	45
3.2 Preliminaries	47
3.3 Summarization Principles	47
3.4 Problem Classification	54
3.5 Variants of Summarization Principles and Problems	62

CONTENTS

3.6	Research Targets for Summarization Problems	68
3.7	A Review of Summarization Algorithms	70
4	Filtering the Clique Explosion: Enumerating ℓ-Isolated Maximal Cliques Efficiently	83
4.1	Introduction	83
4.2	Preliminaries	85
4.3	Finding an ℓ -isolated Maximal Clique is NP-complete	87
4.4	Pruning for ℓ -Isolated Maximal Clique Enumeration	89
4.5	Experimental Setup	94
4.6	Experimental Results	96
5	Conclusions and Open Problems	109
II	Bounded Degree and Outerplanar Induced Subgraphs in Planar Graphs	113
6	Introduction to Large Constrained Induced Subgraphs in Planar Graphs	115
7	Large Induced Subgraphs of Bounded Degree in Outerplanar and Planar Graphs	119
7.1	Introduction	119
7.2	Preliminaries	123
7.3	Induced Subgraphs of Outerplanar Graphs	123
7.4	Induced Subgraphs of Planar Graphs	149
8	Large Induced Outerplanar Subgraphs in 2-Outerplanar Graphs	159
8.1	Preliminaries	160
8.2	On the Proof of Theorem 8.1 by Borradaile, Le and Sherman-Bennett	163
8.3	A New Proof of Theorem 8.1	165
9	Large Induced Outerplanar Subgraphs in Planar 3-Trees	223
9.1	Preliminaries	224
9.2	Lower and Upper Bounds	225
10	Conclusions and Open Problems	235

CONTENTS

Appendices	238
A Other Research Activities	241
A.1 Upward Pointset Embeddings of Planar <i>st</i> -Graphs	243
B Publications	279
Bibliography	281

Introduction

Graphs are fundamental structures for modeling complex systems, ranging from social and biological networks to digital infrastructures and financial systems. Understanding the organization of these networks often requires analyzing their substructures, which can reveal tightly connected groups, recurring patterns, or extremal properties. Among the many approaches, studying *induced subgraphs* has proved both theoretically rich and practically relevant, providing tools to explore network density, connectivity, and combinatorial limits. An induced subgraph preserves faithful information from the original graph: if an edge exists between two nodes in the induced subgraph, it is guaranteed to exist in the original graph, ensuring that no false relationships are introduced. Moreover, since a complete graph admits no non-trivial induced subgraphs other than complete graphs themselves, induced subgraphs are most informative when applied to sparse graphs, a structural setting that closely matches many real-world networks.

A classic task in graph analysis is that of identifying communities, which typically corresponds to partitioning a network into highly dense groups that are weakly connected to the rest of the graph. In this context, cliques represent the strictest form of cohesive substructure. A *clique* is a set of vertices that are pairwise adjacent, and it is said to be *maximal* if no additional vertex of the graph can be included without destroying this property. While finding a maximum clique is a well-known NP-hard [1] problem, W[1]-hard when parameterized by the clique size [2, 3], and hard to approximate within a factor of $n^{1-\epsilon}$ [4, 5], real-world sparse networks often allow for efficient enumeration of all maximal cliques in practice [6, 7, 8, 9, 10, 11, 12]. This motivates the study of how such

INTRODUCTION

tightly connected induced subgraphs can support classical analysis tasks such as community detection. Hence, it is natural to ask the following question:

Question 0.1. *How can tightly connected induced subgraphs, such as maximal cliques, support community detection?*

A second challenge arising when dealing with maximal cliques concerns their abundance and redundancy: they may heavily overlap, and their number can be exponential in the size of the graph [13]. Consequently, understanding how to represent or select a meaningful subset of them becomes a central question. Rather than focusing on individual cliques, one is often interested in producing a *summary* that captures the essential information contained in the full collection. Although this problem can be interpreted in a broad and general manner, our aim is to develop systematic taxonomies that classify principled ways of summarizing maximal cliques concisely and effectively. Hence, we can pose the following question:

Question 0.2. *How can we define a summary of maximal cliques, that is, a concise subset that effectively represents all maximal cliques in the graph?*

An important family of sparse graphs that has been extensively studied is that of *planar graphs*, namely graphs that can be embedded in the plane without edge crossings except at common endpoints. Planar graphs play a central role in many applied contexts [14]. Within this setting, a natural and more combinatorial graph analysis question is the following:

Question 0.3. *How large can structurally constrained induced subgraphs be guaranteed to exist within planar graphs?*

Since planar graphs form a broad class, similar questions can be posed for interesting subclasses, yielding both refined guarantees and deeper structural insights. This perspective is motivated by the fact that induced subgraphs often satisfy stronger structural properties than the original graph, and identifying large such subgraphs can be beneficial in several contexts [15], including visual analysis through simpler graph drawings [14], the design of divide-and-conquer algorithms supported by balanced separators [16, 17, 18], and the development of efficient approximation schemes [19, 20]. Questions concerning the search for large induced subgraphs also connect to some of the most well-known conjectures in graph theory [21, 22, 23], highlighting the fundamental and challenging nature of these problems.

This thesis investigates these themes from both algorithmic and theoretical perspectives. Its overarching goal is to address and answer the questions and the issues outlined above: how dense induced substructures such as maximal cliques can be used to support tasks like community detection; how all maximal cliques can be meaningfully summarized; and how large induced subgraphs with specific structural constraints can be guaranteed to exist in sparse graph families, such as planar graphs.

Through this dual viewpoint, the thesis aims to advance our understanding of induced subgraphs as a unifying tool for analyzing complex networks and sparse combinatorial structures.

The work is organized into two complementary parts. **Part I** (Chapters 2 to 5) addresses overall **Question 0.1** and **Question 0.2**. In particular, it focuses on applied problems in which maximal cliques serve as the primary building blocks to capture dense regions of the graph. Here, the goal is to leverage clique structures to either partition networks into communities or to summarize them in a compact and meaningful way. **Part II** (Chapters 6 to 9) is more theoretical and addresses **Question 0.3**. In particular, it focuses on extremal combinatorial problems: given a family of graphs with n vertices, the aim is to determine how large induced subgraphs can be guaranteed to exist under specific constraints. This line of research is connected to classical problems in graph theory, including well-studied and long-standing conjectures.

Beyond the two main research directions described above, **Appendix A** (**Appendix A.1**) presents additional results obtained during the doctoral work. Although not directly aligned with the central objectives of the dissertation, these contributions address relevant questions in graph drawing and computational geometry and thus complement the overall scientific landscape of the thesis. In particular, **Appendix A** focuses on problems concerning the upward straight-line drawing of directed acyclic planar graphs on prescribed pointsets, as well as on problems related to finding planar Hamiltonian cycles (often called *polygonalizations*) that contain a given set of points.

The thesis is organized as follows.

Chapter 1 introduces the preliminaries and notation used throughout the thesis.

Chapter 2 explores how maximal cliques can guide graph partitioning and community detection. After enumerating all maximal cliques, we encode them in a matrix inspired by term–document representations in natural language processing. By weighting and normalizing this matrix, vertices are embedded into a metric space where clustering algorithms can identify cohesive groups.

This approach yields partitions that match or surpass state-of-the-art community detection methods on a variety of real-world and synthetic networks, demonstrating the power of combining combinatorial structures with AI-driven clustering.

Chapter 3 addresses the challenge of summarizing maximal cliques in large graphs. Since their number may be exponential and they may heavily overlap, we develop a systematic taxonomy of summarization principles and problem variants, unifying existing approaches in a new perspective and introducing new ones. The resulting framework enables the extraction of meaningful subsets of maximal cliques according to size, isolation, coverage, overlap, or qualitative criteria, offering both conceptual clarity and practical tools for graph analysis.

Chapter 4 investigates *isolated maximal cliques*, a notion selecting cliques with limited connectivity to the rest of the graph. We prove that finding such cliques is NP-complete when the isolation threshold is part of the input, ruling out output-sensitive polynomial-time enumeration unless $P = NP$. Motivated by this, we build on the Bron–Kerbosch algorithm with pivoting [6, 10, 24] to design pruning heuristics that restrict the search to promising branches. Experiments on real-world and synthetic graphs show that these heuristics substantially speed up computation compared to existing approaches.

Chapter 5 concludes the first part by summarizing its main contributions and discussing open problems and future research directions related to the use of maximal cliques as a model of induced subgraphs for graph analysis.

Chapter 6 introduces the problem of finding large constrained induced subgraphs in planar graphs and presents a unified framework to model the problems studied in **Part II**.

Chapter 7 studies induced subgraphs of bounded degree in planar and outerplanar graphs. For a given k , we establish lower and upper bounds on the size of induced subgraphs with maximum degree at most k that are guaranteed to exist in any planar or outerplanar graph. Our bounds are tight for outerplanar graphs when $k = 3$.

Chapter 8 focuses on outerplanar induced subgraphs in 2-outerplanar graphs. We consider 2-outerplanar graphs with a fixed embedding (i.e., 2-outerplane graphs) and prove a tight bound on the size of the largest induced subgraph whose inherited embedding is outerplane, that is, an induced subgraph in which all vertices lie on the outer face with respect to the given embedding. This bound was previously established in [15]; here, we identify a major flaw in their proof and recover the result.

Chapter 9 focuses on outerplanar induced subgraphs in planar 3-trees. In this case, we derive both lower and upper bounds on the size of the largest

outerplanar induced subgraph that is guaranteed to exist in any planar 3-tree.

Chapter 10 concludes **Part II** by summarizing its main contributions and discussing open problems and directions for future research related to the study of extremal properties of planar graphs.

Appendix A.1 examines upward pointset embeddings of planar *st*-graphs. It addresses the problem of determining whether a directed acyclic planar graph admits an upward straight-line drawing on a prescribed pointset. In particular, the chapter establishes new NP-hardness results for restricted families of planar *st*-graphs, presents an XP algorithm parameterized by the size of the largest *st*-cutset, provides a polynomial-time algorithm for the case where the graph consists of two internally-disjoint *st*-paths, and introduces a polynomial-time enumeration algorithm to list all upward planar Hamiltonian cycles on a given pointset.

Chapter 1

Preliminaries

In this chapter, we provide the preliminaries and notation used throughout this thesis. We adopt standard terminology from Graph Theory [25], and we use the terms *graph* and *network* interchangeably.

Basic definitions on graphs. Let $G = (V, E)$ be a graph. We denote by $V(G)$ and $E(G)$ its vertex set and edge set, respectively. When clear from the context, we simply use V and E . Since we consider undirected graphs, an edge (u, v) is the same as an edge (v, u) . We say that two vertices $v, v' \in V$ are *adjacent* if they are incident to the same edge, i.e., if $(v, v') \in E$. If $(v, v') \in E$, we also say that v is a *neighbor of v'* . For a vertex $v \in V$, let $\delta(v)$ denote its *degree*, i.e., the number of edges incident to v . We denote by $N(v) = \{u \in V \mid (u, v) \in E\}$ the *neighborhood of v* . Clearly, $|N(v)| = \delta(v)$. The *maximum degree of G* , also simply called the *degree of G* , is the maximum degree over all vertices of G and is denoted by $\Delta(G)$. A *self-loop* in G is an edge $(v, v) \in E$. A set of *multiple edges* in G is a set of edges connecting the same two vertices $u, v \in V$. A graph G is *simple* if it contains neither self-loops nor multiple edges. Given a graph $G' = (V', E')$, we say that G' is a *subgraph of G* if $V' \subseteq V$ and $E' \subseteq E$. For any subset $I \subseteq V$, we denote by $G[I]$ the subgraph of G *induced by I* , that is, the graph whose vertex set is $V(G[I]) = I$ and whose edge set is $E(G[I]) = \{(u, v) \in E \mid u, v \in I\}$. When clear from the context, the vertex set and the edge set of $G[I]$ are denoted by $V[I]$ and $E[I]$, respectively. Throughout the thesis, unless stated otherwise, we consider simple undirected graphs.

Cliques and maximal clique enumeration. A *clique* of G is a complete subgraph of G . A clique C is *maximal* if it is not strictly contained in any other

1. PRELIMINARIES

clique of G , and it is *maximum* if there is no clique in G with more vertices. Given an integer k , deciding whether G contains a clique of size at least k is a classical NP-hard problem [1], which is W[1]-hard when parameterized by the clique size [2, 3] and hard to approximate within a factor of $n^{1-\epsilon}$ [4, 5]. Consequently, computing a maximum clique is intractable in general.

Algorithm 1.1: TomitaTanakaTakahashi

Input: Graph $G(V, E)$

Output: $\mathcal{M}(G)$

```

1 PROC MCE( $\emptyset, V, \emptyset$ )
2 Procedure PROC MCE( $C, P, X$ )
3   if  $P = \emptyset$  and  $X = \emptyset$  then
4     yield  $C$  as a maximal clique;
5     choose a pivot vertex  $v_p$  in  $P \cup X$  with highest  $|N(v_p) \cap P|$ ;
6     foreach  $v \in P \setminus N(v_p)$  do
7       PROC MCE( $C \cup \{v\}, P \cap N(v), X \cap N(v)$ );
8        $P \leftarrow P \setminus \{v\}$ ;
9        $X \leftarrow X \cup \{v\}$ ;
```

Let $\mathcal{M}(G)$ denote the set of all maximal cliques of G . It is known that there exist n -vertex graphs with $3^{n/3}$ maximal cliques [13], hence $|\mathcal{M}(G)|$ can be exponential in n . This implies that enumerating all maximal cliques can be computationally demanding, as their number may grow exponentially with the size of the graph. Nevertheless, several centralized [6, 7, 8, 9, 10, 12] and distributed [26, 27, 28, 29] algorithms have been developed to efficiently enumerate them in real-world graphs. Most known approaches to enumerate maximal cliques are based on the pioneering algorithm proposed by Bron and Kerbosch [6], which was later refined by Tomita, Tanaka, and Takahashi [10, 24]. The approach in [24] plays a central role in this thesis, as it provides the algorithmic framework underlying several of our results; for this reason, we describe it in detail. The approach is based on a recursive backtracking procedure outlined in Algorithm 1.1. Three sets of vertices are maintained in the procedure: C , P , and X . Set C represents the current clique that is additively constructed; P is the set of candidate vertices that can potentially be added to C ; and X is the set of vertices that should not be considered as they have already been included in a previously reported maximal clique from the current clique C or from a subset of it. Vertices in P and X are neighbors of all vertices in C . We

1. PRELIMINARIES

refer to P and X as the perspective set and the taboo set, respectively. The main idea is to add a vertex from the set of candidate vertices in P to expand the current clique C , until it becomes maximal.

Algorithm 1.1 invokes Procedure $\text{PROC MCE}(C, P, X)$ with $C = \emptyset$, $P = V$, and $X = \emptyset$ (Line 1). First, the procedure checks whether both P and X are empty. If so, C cannot be further expanded and is reported as a maximal clique (Line 4). Otherwise, in order to reduce the number of recursive calls, a pivot vertex v_p is chosen (Line 5) and the procedure is recursively called (Line 7) for each vertex $v \in P$ that is not a neighbor of v_p (this is because any maximal clique containing $C \cup \{w\}$, where w is a neighbor of v_p , can be reported either by recurring on $C \cup \{v_p\}$ or by recurring on $C \cup \{u\}$, where $u \in P$ is not a neighbor of v_p). The recursion is performed with $C = C \cup \{v\}$, $P = P \cap N(v)$, and $X = X \cap N(v)$ to ensure that every vertex in P and X is a neighbor of all vertices in C . When the recursive call returns, v is moved from P to X (Lines 8 and 9).

When invoked on a graph G , **Algorithm 1.1** implicitly explores a search tree $\mathcal{T}_G$. Each node of $\mathcal{T}_G$ corresponds to a recursive call and is uniquely associated with a triple (C, P, X) . The leaves of $\mathcal{T}_G$ are nodes of the form $(C, \emptyset, X)$. If $X = \emptyset$, the leaf corresponds to a maximal clique that is reported. Otherwise, the maximality condition is not satisfied (vertices in X could be added to C), the clique is discarded, but no recursive call needs to be done.

Consider as an example the graph depicted in **Figure 1.1a**. When Procedure PROC MCE (**Algorithm 1.1**) is launched for the first time, $C = \emptyset$, $P = V$, and $X = \emptyset$ (all vertices of **Figure 1.1b** are green). Since vertex v_3 satisfies the condition of Line 5, it is chosen as the pivot vertex v_p . Due to Line 6, the procedure will be recursively called for each $v \in P \setminus N(v_3)$, i.e., for $v = v_3$ only. Hence, the second recursive call of PROC MCE has $C = \{v_3\}$, $P = \{v_1, v_2, v_4\}$, and $X = \emptyset$ (see **Figure 1.1c**). Assuming that vertex v_1 is now chosen as the pivot vertex v_p (notice that v_2 could be another possibility), the procedure will be recursively called for each $v \in P \setminus N(v_1)$, i.e., for $v = v_1$ and $v = v_4$. The third recursive call has $C = \{v_1, v_3\}$, $P = \{v_2\}$, and $X = \emptyset$ (see **Figure 1.1d**). Vertex v_2 is then chosen as the pivot vertex and the procedure will be recursively called for $v = v_2$ only. Therefore, the fourth call of PROC MCE has $C = \{v_1, v_2, v_3\}$, $P = \emptyset$, and $X = \emptyset$ (see **Figure 1.1e**). At this point, the conditions of Line 3 are met and the maximal clique $\{v_1, v_2, v_3\}$ is produced. Upon termination of the fourth call of PROC MCE , Lines 8 and 9 of the third recursive call move v_2 from P into X (see **Figure 1.1f**). This terminates also the third call. Similarly, Lines 8 and 9 of the second recursive call move v_1 from P to X (see **Figure 1.1g**). Line 6 of the second call iterates with $v = v_4$ and recursively calls PROC MCE (Line 7)

1. PRELIMINARIES

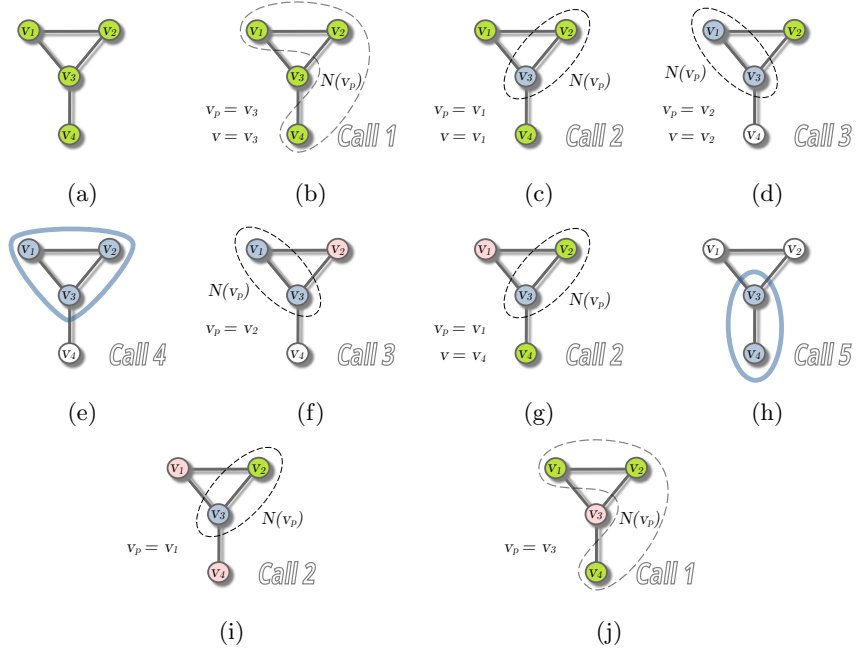


Figure 1.1: A running example of Algorithm 1.1 on a small graph. Green, blue, and red vertices belong to P , C , and X , respectively.

with $C = \{v_3, v_4\}$, $P = \emptyset$, and $X = \emptyset$ (see Figure 1.1h). The conditions of Line 3 are met again and the maximal clique $\{v_3, v_4\}$ is produced. When the fifth call terminates, Lines 8 and 9 of the second call move v_4 from P to X . The second call also terminates and Lines 8 and 9 of the first call move v_3 from P to X . This terminates the first call, too. In the end, Algorithm 1.1 produced the maximal cliques $\{v_1, v_2, v_3\}$ and $\{v_3, v_4\}$.

Figure 1.2 shows all the cliques of the graph depicted in Figure 1.1a and the directed acyclic graph (DAG) of their inclusion relationships. The source of the DAG corresponds to the empty set while the maximal cliques are its two sinks. The nodes of the DAG with a solid border are those on which Algorithm 1.1 has been launched, exploring only a restricted portion of the solution space. The numbers close to these nodes correspond to the five recursive calls.

The worst-case running time of Algorithm 1.1 is $\mathcal{O}(3^{n/3})$ [24], matching the number of maximal cliques in Moon–Moser graphs [13]. In [30] it is also proved

1. PRELIMINARIES

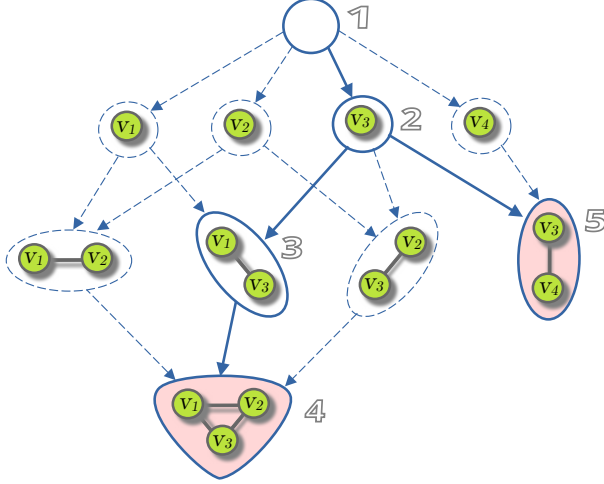


Figure 1.2: The inclusion relationship of the cliques of the small graph depicted in Figure 1.1. The maximal cliques are the sinks of the DAG. The nodes of the DAG with solid border have been explored by Algorithm 1.1.

that Algorithm 1.1 has $\Omega(3^{n/6})$ delay and that, even changing the pivoting strategy, a variant having polynomial delay cannot be designed unless $P = NP$. Notwithstanding these asymptotic complexity results, it has been experimentally observed that approaches based on Algorithm 1.1 are fast in practice [8, 24]. FPT enumerative approaches also exist, either in the degeneracy [8] or in the c -closedness [31, 32] of the graph¹.

Planar graphs. We use standard terminology in Graph Drawing [14]. A *drawing* of a graph maps each vertex to a point in the plane and each edge to a Jordan arc between the points representing the end-vertices of the edge. A drawing is *planar* if no two edges cross, except at common end-vertices. A *planar graph* is a graph that admits a planar drawing. We refer to the topological information associated with such a drawing as a *plane embedding*, and we call a graph together with a plane embedding a *plane graph*. A *maximal planar graph* is a planar graph such that adding any edge to it results in a graph that contains a multiple edge or is not planar. A planar drawing partitions the plane into connected regions, called *faces*; the unique unbounded face is the *outer face*,

¹The degeneracy, a measure of sparseness, is the largest k such that the graph contains a k -core. A graph is c -closed if every pair of vertices that share at least c neighbors is adjacent.

1. PRELIMINARIES

while the other faces are *internal*. A planar drawing is *outerplanar* if every vertex is incident to the outer face. An *outerplanar graph* is a graph that admits an outerplanar drawing. A *maximal outerplanar graph* is an outerplanar graph such that adding any edge to it results in a graph that contains a multiple edge or is not outerplanar. Consider a maximal outerplanar graph and observe that it has a unique outerplanar drawing, up to a reflection and a homeomorphism of the plane. We call *outerplane embedding* the topological information associated with such a drawing. Also, we call a graph together with an outerplane embedding an *outerplane graph*. The *weak dual* of a plane graph G is the graph with a vertex for each internal face of G and with an edge between two vertices if the corresponding faces of G share an edge.

Part I

Maximal Cliques as Induced Subgraphs

High-Modularity Graph Partitioning Through NLP Techniques and Maximal Clique Enumeration

2.1 Introduction

Community detection is a fundamental tool in network analytics, with widespread applications across various domains, including social networks, biology and bioinformatics, Internet and Web, recommendation systems, cybersecurity, transportation and infrastructure networks, financial networks, neuroscience, and criminal networks. Intuitively, a community is a subset of vertices that are tightly connected among them. When every possible pair of vertices are adjacent, the community is fully connected and can be modeled by a clique. Graph-theoretic properties based on cliques, such as k -plexes [33, 34, 35, 36], s -defective [37] or γ -quasi cliques [38] yield the strictest definitions of community. We refer to [39] for a survey. However, the number of graph-theoretic communities is usually huge and their overlap is too large to allow the user to exploit such communities to analyze and decompose the network.

In contrast to graph-theoretic approaches, a more intuitive definition of community for users is that of partitioning, where the vertex set is divided into distinct disjoint blocks. Measures such as modularity capture the quality of the partition (we refer to [Section 2.2](#) for a formal definition). Unfortunately, computing partitions with maximum modularity is NP-complete [40] and exact

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

algorithms are not feasible, even for sparse networks. As a consequence, most existing approaches to network partitioning rely on heuristics.

Chapter contributions. In this chapter¹, we aim to establish a connection between communities defined in a graph-theoretic sense and vertex partitions. While the former can be computed using exact algorithms, the calculation of the latter relies on heuristic methods.

By using graph-theoretic communities as a guide, our approach applies an AI-driven method to the vertex partitioning problem. In more details, we begin by encoding maximal cliques into a matrix, with rows representing vertices and columns corresponding to cliques, akin to term-document matrices used in NLP domain. Through a weighting operation performed on this matrix, similar to function TF-IDF, we then assign higher values to maximal cliques that are large and loosely connected to the rest of the network.

This matrix, once normalized, can be viewed as an embedding of the graph vertices into a metric space. Finally, we cluster the rows of the matrix (i.e., the points in the metric space) according to their similarity (i.e., distance) in order to partition the network.

This approach proves beneficial both when the number k of blocks of the partition is specified in advance, and when such a number is not specified. For the first scenario, we produced two algorithms, dubbed k -AGGL-CLIQUE-TF-IDF and k -MEANS-CLIQUE-TF-IDF, that use an agglomerative approach [43] or the k -Means algorithm [44] to cluster the matrix rows, respectively. In Section 2.5 we experimentally show that the two algorithms outperform the well-known METIS graph partitioning algorithm [45] in terms of quality of the produced partitions, measured both with respect to absolute partitioning metrics and with respect to the ground truth, at the cost of an increased computation time.

When the number k of blocks of the partition is not provided in advance, building upon preliminary results presented in [41], we add a further step to the approach in order to find a good value for k . The obtained algorithm, named CLIQUE-TF-IDF, is experimentally compared with several state-of-the-art approaches in the literature considering both real-world and synthetic networks. The experiments reported in Section 2.6 show that CLIQUE-TF-IDF is among the approaches with the best performance, although the computation times are

¹An earlier and shorter version of some of the results in this chapter appeared in [41]. The present chapter significantly extends that work; it is based on joint work with Irene Finocchi and Maurizio Patrignani and has also appeared as a technical report in [42].

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

longer than those of Leiden [46], which is a good algorithm both with respect to effectiveness and with respect to efficiency.

In order to validate the use of maximal cliques of a graph to obtain an embedding of its vertices into a metric space for clustering purposes, which is proposed for the first time in this chapter, we also performed experiments where this embedding phase is replaced by alternative state-of-the-art graph embedding approaches. Section 2.7 shows that the adopted embedding phase is more effective, more efficient, and also more practical, as it does not require many tuning parameters as is the case for alternative approaches.

As an additional contribution, in Section 2.4 we also show how to generate a benchmark of networks whose ground truth has the same modularity independently of their size. This result can be of interest in its own right.

Related work. The graph partitioning problem has attracted significant interest from both theoretical and practical perspectives and has been extensively studied. Various problem variants have been considered in the literature, with some works employing AI techniques to tackle different issues. For instance, when vertices have associated attributes that must be considered in the partitioning, contrastive learning and graph neural networks facilitate the learning of clusterable features, resulting in partitions that align well with the ground truth [47]. Algorithms extending traditional approaches for partitioning attributed graphs are discussed in [48].

In correlation partitioning, the edges are weighted quantitatively, representing vertex similarity. In this case, the objective is to maximize a measure that typically increases when edges with negative weights span across different partitions. This problem has been studied in [49, 50], and an online version has been studied in [51].

Another variant is structural clustering, where vertices in the same partition blocks are expected to have similar “roles” within the graph. In this context, graph neural networks have been applied to vertex classification and graph classification tasks [52, 53, 54]. Additionally, structural vertex embedding techniques can be applied to identify an effective partition [55].

Differently from the aforementioned works, the focus of this chapter is on positional clustering for graphs that have no attributes: vertex similarity depends on adjacency, and the partition quality is evaluated using modularity. This problem has been addressed in the literature in many previous works that present a variety of combinatorial algorithms, both when the number k of desired partition blocks is known in advance and when it is not. For the former case, the well-known METIS approach [45] can be applied in either its recursive bisection

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

or multiway partitioning variants. A non-negative matrix factorization-based algorithm is detailed in [56].

The case where k is not part of the input has attracted more attention in the literature. Agglomerative hierarchical methods that maximize modularity measure are discussed in [57, 58, 46]. Other approaches combine random walks on the graph with a similarity matrix [59] or use information theory methods [60]. Label propagation techniques are considered in [61], while statistical physics models are used in [62]. Additional heuristic approaches, which generally show lower effectiveness, are discussed in [63, 64, 65].

Finally, vertex embeddings based on positional information can also be leveraged to compute a partitioning [66, 67, 54], like in the case of structural clustering. For a comprehensive survey on vertex embedding techniques, particularly those leveraging graph neural network models, see [68]. When vertices have associated attributes, embedding methods are further detailed in [69].

Chapter organization. The rest of the chapter is organized as follows. In [Section 2.2](#) we introduce notation and provide preliminary background. Our approach is described in detail in [Section 2.3](#). The experimental setup is provided in [Section 2.4](#), while [Sections 2.5 to 2.7](#) discuss the experimental analyses. CLIQUE-TF-IDF is compared to state-of-the-art methods in terms of both efficiency and effectiveness, focusing on different scenarios: when the number of communities is known in advance, when it is not known, and when only the embedding phase is considered.

2.2 Preliminaries

We refer to [Chapter 1](#) for basic definitions on graphs. Let $G = (V, E)$ be a simple undirected graph, and let n denote the number of vertices of G .

Let A be the adjacency matrix of G , where $A_{uv} = 1$ if vertex u and vertex v are adjacent, and 0 otherwise. In an undirected graph, A is symmetric, so $A_{uv} = A_{vu}$. A [partition of \$G\$](#) is a subdivision of V into distinct blocks $\{V_1, \dots, V_k\}$, such that $\bigcup_{i=1}^k V_i = V$ and $V_i \cap V_j = \emptyset$ for every $i \neq j$. Each block is intended to represent an internally dense area that is weakly connected to the other vertices.

Modularity. [Modularity](#) can be used to measure the quality of graph partitions and is defined in [57] as follows:

$$Q = \frac{1}{2m} \sum_{i=1}^k \sum_{u,v \in V_i} \left(A_{uv} - \frac{\delta_u \delta_v}{2m} \right) \quad (2.1)$$

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

Here $\frac{\delta_u \delta_v}{2m}$ denotes the probability that an edge exists between vertices u and v in a random network model that maintains the vertices' degree distribution. The modularity measure Q ranges in the interval $[-\frac{1}{2}, 1]$, where non-zero values indicate deviations from randomness, with higher values suggesting a stronger community structure.

In spite of the fact that maximizing modularity is NP-hard [40], this measure is widely used in previous works to assess the quality of communities, particularly in cases where a ground truth is unavailable or requires validation [58, 57, 46].

Permanence. This is a more recent quality measure for graph partitions introduced in [70]. We first need some preliminary notation:

- $b(v)$ is the block of the partition v belongs to;
- $I(v)$ is the number of edges incident to v and internal to $b(v)$;
- $E_{max}(v)$ is the maximum number of edges of v to a single block of the partition different from $b(v)$;
- $c_{in}(v)$ is the internal clustering coefficient of $b(v)$, i.e., the ratio of the existing edges and the total number of possible edges among the neighbors of v that are part of $b(v)$.

Permanence is defined as $P(G) = \frac{1}{n} \sum_{v \in V} P(v)$ where:

$$P(v) = \left(\frac{I(v)}{\max\{1, E_{max}(v)\}} \cdot \frac{1}{\delta_v} \right) - (1 - c_{in}(v)) \quad (2.2)$$

Observe that $P(v) \in [-1, 1]$. The maximum value of $P(v)$ is equal to $c_{in}(v) = 1$ and is obtained when $b(v)$ is an isolated clique of G (in this case $I(v) = \delta(v)$). On the other side, $P(v)$ is close to -1 when $I(v) \ll \delta(v)$ and $c_{in}(v) = 0$. Consequently, also $P(G) \in [-1, 1]$.

TF-IDF. The term frequency-inverse document frequency (in short, TF-IDF) measures the importance of terms in documents from a collection $D = \{d_1, \dots, d_r\}$ of size r . We assume terms to be taken from a set $T = \{t_1, \dots, t_s\}$ of size s , and denote by $\tau_{t,d}$ the number of occurrences of a term $t \in T$ within a document $d \in D$. Since some terms may be more frequent in general, we count in δ_t the number of documents containing t and define the *inverse document frequency* of t as $\gamma_t = \log(\frac{|D|}{\delta_t})$. TF-IDF is a weighting function defined as $\omega_{t,d} = \tau_{t,d} \cdot \gamma_t$ [71]: its values are higher for terms appearing in a small subset of documents, and lower when a term is not common within a document or is common in the entire corpus.

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

Hierarchical clustering. Given a set $P = \{p_1, \dots, p_n\}$ of n points within a d -dimensional space and an integer k , the objective of geometric clustering is to divide P into k non-empty clusters $\{P_1, \dots, P_k\}$ such that $\bigcup_{i=1}^k P_i = P$ and $P_i \cap P_j = \emptyset$ for every $i \neq j$. A well-known solution is based on the k -Means algorithm [44].

Another well-known method to compute a geometric clustering of P is hierarchical clustering, valued for its flexibility in identifying both local and global structures. One of the most popular variants is called *agglomerative clustering* [43] and begins by treating each individual point as a separate cluster, successively merging them into larger clusters. This approach relies on three key components: (i) a metric for calculating distances between points, (ii) a “linkage criterion” to measure distances or similarities between clusters to determine which to merge, and (iii) a “stopping criterion” to end the computation at an appropriate hierarchical level. Unlike other clustering algorithms based on density or centroids, hierarchical approaches do not require any initialization. Moreover, it benefits from only using distances, rather than absolute coordinates, making it more broadly applicable.

2.3 Overview of the Approach

This section introduces our graph partitioning method. We leverage the efficient enumeration of the maximal cliques of a graph G and follow a pipeline consisting of the following macro-steps:

VERTEX-COMMUNITY MATRIX: after computing all the maximal cliques, we construct a vertex-community matrix Z , with values reflecting the sizes of the cliques.

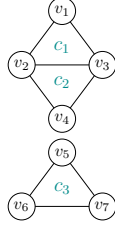
TF-IDF TRANSFORMATION: a weighting operation similar to TF-IDF is applied to Z , assigning higher values to maximal cliques that are both large and weakly connected to the rest of the network.

CLUSTERING: by regarding the matrix rows as a set of points embedded in a multi-dimensional space, the graph partition is ultimately determined by clustering these points.

2.3.1 Vertex-community Matrix

To enumerate maximal cliques, we use the algorithm introduced in [6] with the pivoting strategy discussed in [10, 24] (see [Algorithm 1.1](#)). We denote by

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION



(a) Graph G

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	3	3	3	0	0	0	0
v_2	3	6	6	3	0	0	0
v_3	3	6	6	3	0	0	0
v_4	0	3	3	3	0	0	0
v_5	0	0	0	0	3	3	3
v_6	0	0	0	0	3	3	3
v_7	0	0	0	0	3	3	3

(b) Co-participation matrix X

	c_1	c_2	c_3
v_1	1	0	0
v_2	1	1	0
v_3	1	1	0
v_4	0	1	0
v_5	0	0	1
v_6	0	0	1
v_7	0	0	1

(c) Clique-incidence matrix Y

	c_1	c_2	c_3
v_1	9	6	0
v_2	15	15	0
v_3	15	15	0
v_4	6	9	0
v_5	0	0	9
v_6	0	0	9
v_7	0	0	9

(d) Vertex-community matrix Z

Figure 2.1: (a) An example graph G . The matrix Z (d) is the product of X (c) and Y (b).

$C = \{c_1, \dots, c_d\}$ the set of all maximal cliques. Let n and d be the number of vertices and the number of maximal cliques of G , respectively. We use two auxiliary data structures.

Clique-incidence matrix Y . Vertex containment in maximal cliques is described by a *clique-incidence* matrix $Y \in \mathbb{N}^{n \times d}$. More formally:

$$Y_{i\ell} = \begin{cases} 1 & \text{if vertex } v_i \text{ belongs to maximal clique } c_\ell \\ 0 & \text{otherwise} \end{cases}$$

As an example, the incidence matrix Y depicted in Figure 2.1(c) shows clique membership for the graph of Figure 2.1(a), which contains three maximal cliques c_1 , c_2 , and c_3 . Our goal is to cluster the rows of Y based on their similarity, grouping in the same partition vertices belonging to very similar sets of maximal cliques. For instance, in Figure 2.1(c), vertices v_1 and v_2 share more similarities compared to vertices v_1 and v_5 .

Co-participation matrix X . Directly using the matrix Y without any post-processing would not emphasize whether the vertices belong to maximal cliques

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

of large or small size. Also, we would like that two vertices that do not fall into the same maximal clique (see, for example, vertices v_1 and v_4 of graph G in [Figure 2.1\(a\)](#)) had some similarity in matrix Y based on the fact that they have neighbors in common in some other maximal clique (see, for example, the rows 1 and 4 in [Figure 2.1\(b\)](#) corresponding to vertices v_1 and v_4).

In order to solve the above two problems, we introduce an auxiliary matrix $X \in \mathbb{R}^{n \times n}$, called *co-participation* matrix. Each entry X_{ij} of matrix X accounts for the sizes of the maximal cliques to which vertices v_i and v_j both belong. Precisely, let w_ℓ be the number of edges of a maximal clique c_ℓ . For any pair of vertices $\langle v_i, v_j \rangle$, where $i, j = 1, \dots, n$ and $i \neq j$, we set $X_{ij} = \sum_{v_i, v_j \in c_\ell} w_\ell$. As for the diagonal of matrix X , we set $X_{ii} = \sum_{v_i \in c_\ell} w_\ell$. Observe that matrix X is symmetric. For the example graph of [Figure 2.1\(a\)](#), since both vertices v_2 and v_3 share the maximal clique c_1 , with $w_1 = 3$, and the maximal clique c_2 , with $w_2 = 3$, we have $X_{23} = w_1 + w_2 = 6$.

Vertex-community matrix Z . We use matrix X to process matrix Y performing the matrix multiplication $Z = X \cdot Y$. We call the obtained matrix Z *vertex-community* matrix. Observe that Z has the same size $\mathbb{R}^{n \times d}$ as matrix Y . Each entry $Z_{i\ell}$ of Z can be viewed as the sum of two types of terms:

- *Actual involvement* of vertex v_i into the clique c_ℓ : when $v_i \in c_\ell$ we sum to $Z_{i\ell}$ the value $|c_\ell| \cdot w_\ell$ (see for example Z_{42} in [Figure 2.1\(d\)](#), where $Z_{42} = |c_2| \cdot w_2 = 3 \times 3 = 9$). This value quadratically increases with the number of vertices of c_ℓ .
- *Transitive involvement* of vertex v_i into the clique c_ℓ : when $v_i \notin c_\ell$, for each vertex $v_j \in c_\ell$ we sum to $Z_{i\ell}$ the term $w_{\ell'}$, where $c_{\ell'}$ is a maximal clique to which both v_i and v_j belong. For example, vertex v_4 has a transitive involvement in clique c_1 because of its neighbors v_2 and v_3 . Hence, value $Z_{41} = 6$ is the sum of the two terms $X_{42} \times Y_{21} = w_2 = 3$ and $X_{43} \times Y_{31} = w_2 = 3$.

2.3.2 TF-IDF Transformation

Matrix Z can be compared to the term-frequency matrix used in Information Retrieval, where occurrences of terms in documents are encoded. In our case, vertices correspond to documents and maximal cliques correspond to terms. Following this analogy, we aim to assign more importance to a maximal clique (term) for a vertex (document) when the corresponding column in Z is sparse (i.e., the term is infrequent). This is because a maximal clique that is weakly

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

connected to the rest of the graph is less desirable to split across different blocks of the partition. We therefore compute a vector $\Gamma = [\gamma_1, \dots, \gamma_d] \in \mathbb{R}^d$, where, for each $\ell = 1, \dots, d$, $\gamma_\ell = \log(\frac{n}{|\delta_\ell|})$, where δ_ℓ is the number of non-zero elements of column ℓ of Z , analogous to the inverse document frequency. We then modify Z by performing, for each $i = 1, \dots, n$, an element-wise multiplication of row Z_i with Γ , i.e., Z_i is replaced by $Z_i \odot \Gamma$, where $\odot$ denotes the Hadamard product between two vectors. In the example of [Figure 2.1](#), after the transformation described above, the value for the element $Z_{1,2}$ for clique c_2 with respect to vertex v_1 is $\omega_{1,2} = \tau_{1,2} \cdot \gamma_1 = Z_{1,2} \cdot \log(\frac{n}{|\delta_2|}) = 6 \cdot 0.8 = 4.8$.

Finally, to obtain an embedding for the vertices, we normalize each row of Z so that each vertex corresponds to a unit-length vector.

2.3.3 Clustering

Matrix Z can be viewed as a set of n points (the vertices in V) into a d -dimensional space (the cliques). Hence, we can perform geometric clustering of the graph by partitioning these n points into a specified number k of clusters. To achieve this, we apply two alternative methods: (i) an agglomerative clustering approach [\[43\]](#), which initially assigns each point to its own cluster and then iteratively merges the two closest clusters, where the distance between clusters is defined as the average distance between their vertices. This process continues until the desired number of clusters, k , is reached. We call the obtained algorithm k -AGGL-CLIQUE-TF-IDF; (ii) k -Means algorithm [\[44\]](#), a renowned iterative centroid-based approach to produce k -way partitions of points embedded into a metric space. We call the obtained algorithm k -MEANS-CLIQUE-TF-IDF.

Choosing the number of clusters. Since the algorithm requires the number of clusters k as input, we employ a straightforward heuristic by performing a binary search over possible k values to find one that achieves a high modularity score.

In fact, we experimentally assessed that the modularity of the partition computed on matrix Z is usually very close to a bitonic curve with respect to number k of blocks of the partition. This empirical property ensures the effectiveness of the binary search. We consider only agglomerative clustering in this scenario. The efficiency of this search procedure depends indeed by two facts: first, the agglomerative clustering hierarchy is computed once at the start and reused to generate k -way partitions for different values of k ; second, when computing the agglomerative hierarchy only the distances among points

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

are used, instead of their actual coordinates. We call the obtained algorithm CLIQUE-TF-IDF.

2.4 Experimental Setup

In this section we discuss quality metrics (Section 2.4.1), dataset generation (Section 2.4.2), and algorithms under evaluation (Section 2.4.3). All experiments were conducted on a laptop with an Intel® Core™ i5-1135G7 CPU (4 cores, 4.2 GHz) and 8 GB of RAM. The selection of parameters in the generation of synthetic networks, as discussed in Section 2.4.2, is especially interesting and can be regarded as an additional contribution of the chapter.

2.4.1 Metrics

In order to compare the effectiveness of the partitioning algorithms we contrast them with respect to the modularity and the permanence of the produced partitions. For the synthetic instances, where a ground truth is also available, we also measure the normalized mutual information (NMI) [71].

2.4.2 Datasets

In our experiments we used two kinds of datasets: real-world networks and synthetic networks. Real-world networks offer a measure of the performances of the algorithms in application contexts. Synthetic networks allow us to explore in a more systematic way the space of the possible inputs. Further, they allow us to compare the output partitioning with the ground truth used to produce the input instances, yielding an additional quality measure.

Real-world networks. We considered a dataset with ten real-world networks of varying sizes, ranging from tens to ten thousand vertices, collected from open repositories [72, 73, 74], which are commonly used as benchmarks in this field. Their characteristics, including the number of maximal cliques, are listed in Table 2.1. Entries marked with * correspond to the largest component of networks, as certain algorithms, such as Pott [62], require connected networks. These components usually include a significant portion of the vertices. Additionally, we pre-processed the networks by removing loops and multiple edges. Networks originally containing loops and/or multiple edges are marked with ** in Table 2.1. The values reported in Table 2.1 refer to the networks after the operations described above.

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

Table 2.1: Real-world networks used in the experimental analysis. Entries labeled with * only refer to the giant component of the networks. Entries labeled with ** refer to networks where loops and multiple edges have been removed.

Network	Alias	vertices	Edges	# Maximal cliques
arenas-email	arenas	1133	5451	3267
arxiv-grqc*	grqc	4158	13422	3385
arxiv-hepth*	hepth	8638	24806	9357
arxiv-hepph*	hepph	11204	117619	14588
citeseer**	cite	2120	3679	2722
email-eu-core**	email	986	16064	42709
feather-lastfm-social	lastfm	7624	27806	17957
p2p-gnutella04	p2p	10876	39994	38497
sociopatterns-infectious	socio	410	2765	1247
zacharys-karate	karate	34	78	36

Synthetic networks. We used the LFR [75] library to produce synthetic networks with *a priori* known communities where both the vertex degree and the community size distribution are power laws. This library is specifically aimed at comparing different methods for community detection and graph partitioning. Our purpose is that of producing a benchmark of networks to show how some parameters of the instances impact on the performances of the algorithms. In particular, we selected the following parameters:

- Number n of vertices, which is a standard parameter to measure the time complexity of network algorithms.
- Average vertex degree $\langle d \rangle$: this is related to the number of edges in the network, which is a second standard parameter considered in network classification.
- Maximum vertex degree d_{max} : this is a measure of how much evenly the edges are distributed in the network. Scale-free networks have a maximum vertex degree much higher than the average vertex degree and often proportional to the number n of vertices.
- Noise μ (also referred to as “mixing parameter”): this is related to the community structure and determines of how much the internal links of each community are rewired towards vertices of other communities. If this

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

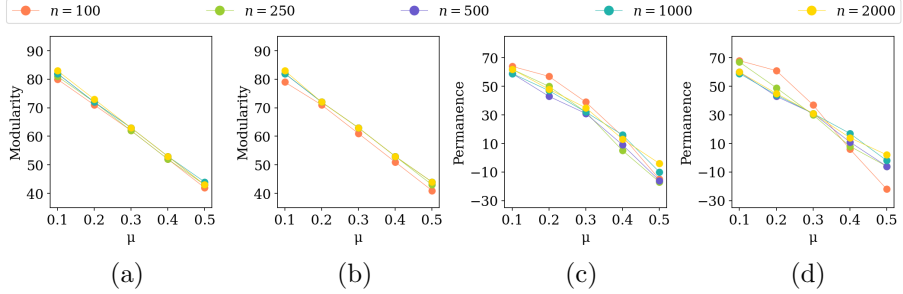


Figure 2.2: The modularity with respect to the ground truth of the synthetic networks produced with the LFR library linearly depends on μ while is independent from n . Similarly, the permanence also exhibits a linear dependency on μ while is substantially independent from n . (a) Modularity plot with $\alpha = 1/10$ and $\beta = 30$. (b) Modularity plot with $\alpha = 1/10$ and $\beta = 40$. (c) Permanence plot with $\alpha = 1/10$ and $\beta = 30$. (d) Permanence plot with $\alpha = 1/10$ and $\beta = 40$.

parameter were set to 1, no links would survive within the community, contradicting our intuitive notion of community.

The LFR library guarantees a scale-free distribution by choosing the number of communities of the network and their sizes. By varying parameters n , $\langle d \rangle$, d_{max} , and μ linearly in their allowed intervals, several combinations do not correspond to feasible networks and the LFR library does not produce any instance. On the other hand, instances produced by the LFR library but having a low value of modularity with respect to the ground truth are not suitable for an experimental analysis of partitioning algorithms: the ground truth becomes challenging, if not impossible, to retrieve in such instances, primarily due to the overwhelming noise that obscures it. With the purpose of obtaining a benchmark of synthetic scale-free networks featuring significant communities we used the following values for the chosen parameters.

- Number of vertices $n \in \{100, 250, 500, 1000, 2000\}$
- Maximum vertex degree $d_{max} = \alpha \cdot n$ with $\alpha \in \{\frac{1}{20}, \frac{1}{15}, \frac{1}{10}, \frac{1}{5}, \frac{1}{3}\}$.
- Average vertex degree $\langle d \rangle = \beta \cdot d_{max} \cdot \frac{\log_{10}(n)}{n}$, with $\beta \in \{30, 35, 40\}$.
Observe that, since $d_{max} = \alpha \cdot n$, this corresponds to having $\langle d \rangle \sim \log(n)$.

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

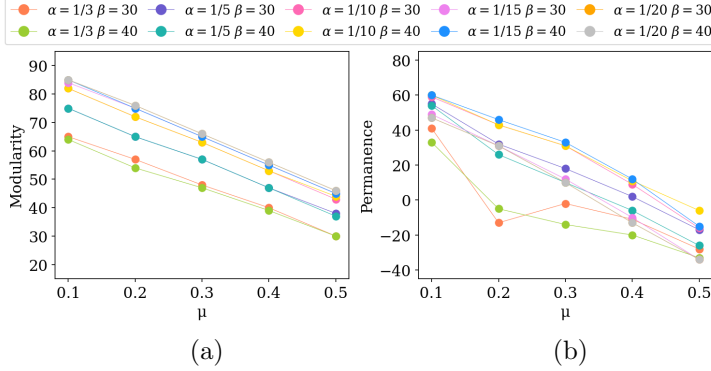


Figure 2.3: The modularity and permanence with respect to the ground truth of the synthetic networks produced with the LFR library for different values of α and β . (a) Plot of modularity for $n = 500$. The four curves for $\alpha \in \{1/15, 1/20\}$ are all overlapped at the top of the plot. (b) Plot of permanence for $n = 500$.

For example, when $\alpha = 1/20$ and $\beta = 30$, then $\langle d \rangle$ varies from 3 ($n = 100$) to 5 ($n = 2000$). Instead, when $\alpha = 1/3$ and $\beta = 40$, then $\langle d \rangle$ varies from 26 ($n = 100$) to 44 ($n = 2000$).

- Noise $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

This choice of the parameters yields networks of guaranteed modularity with respect to the ground truth’s communities. In particular, as it can be seen from Figure 2.2(a)-(b), the modularity of the produced instances for a given pair $\{\alpha, \beta\}$ only depends linearly on μ and is independent of the size n of the graph. The measure of the permanence with respect to the ground truth also seems to show a similar linear dependency on μ and a substantial independence from n (see Figure 2.2(c)-(d)).

Figure 2.3(a) shows that the modularity of the produced instances with respect to the ground truth is independent on the values of β (i.e., from the average degree $\langle d \rangle$). Instead, the modularity decreases when α increases (i.e., when the scale-freeness of the produced networks increases). The measure of permanence (see Figure 2.3(b)) shows a similar descending trend with respect to mixing parameter μ , although more affected by noise.

For each combination of the parameters, we produced five different graph instances and calculated the average value of the metrics considered.

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

2.4.3 Algorithms

We evaluated our novel approach by comparing it with several popular algorithms for network partitioning. In particular, when the number of desired blocks is given, we compared in [Section 2.5](#) with METIS [45]. When the number of blocks of the partition is not known *a priori*, we compared in [Section 2.6](#) with Walktrap [59], CNM [57], Infomap [60], LP [61], Pott [62], and Leiden [46], which is typically preferred over Louvain [58] due to its improved efficiency and partition quality. We were unable to compare with [47] as the code is not yet publicly available.

As alternative network embedding approaches in the initial phase of CLIQUE-TF-IDF we considered in the experimental analysis of [Section 2.7](#) algorithms Node2Vec [66] and DeepWalk [67]. The source code of our implementation can be found at <https://github.com/mdelia17/clique-tf-idf>.

2.5 Evaluation of Clique-TF-IDF for Computing k -way Partitions

Assuming that the number k of desired blocks of the partition is fixed, we evaluated the two variants of our approach called k -AGGL-CLIQUE-TF-IDF and k -MEANS-CLIQUE-TF-IDF (see [Section 2.3](#)) against algorithm METIS using multilevel multiway partitioning (METIS-M) and recursive bisection (METIS-R) [45]. A comparison on real networks is not possible, since for most of them the ground-truth – and hence the value of k – is not known in advance. We therefore focus on the synthetic benchmarks (see [Section 2.4.2](#)).

Varying community interconnectedness (mixing parameter μ). We first analyze how the algorithms are affected by different values of the mixing parameter μ . We fixed $\alpha = 1/10$ and $\beta = 40$. In [Figure 2.4\(a\)-\(c\)](#), when $n = 500$, it can be seen that both our approaches lead in general to a better effectiveness in terms of modularity, permanence and NMI for all the values of μ . Moreover, k -AGGL-CLIQUE-TF-IDF has higher values of modularity and NMI with respect to k -MEANS-CLIQUE-TF-IDF when $\mu > 0.3$. In terms of efficiency, both METIS-M and METIS-R are always more efficient and k -MEANS-CLIQUE-TF-IDF is one order of magnitude slower than k -AGGL-CLIQUE-TF-IDF (see [Figure 2.4\(d\)](#)). Furthermore, the running time of k -MEANS-CLIQUE-TF-IDF slightly increases with μ , while all the other algorithms do not exhibit a running time dependency on the noise μ .

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

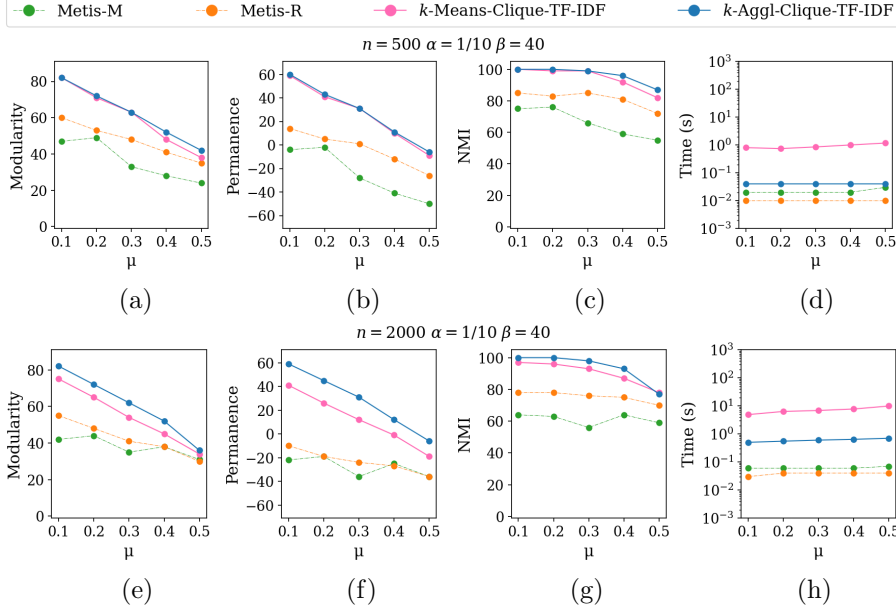


Figure 2.4: Evaluation metrics on synthetic networks when $\alpha = 1/10$ and $\beta = 40$, varying μ , for $n \in \{500, 2000\}$ (rows).

When $n = 2000$, *k*-AGGL-CLIQUE-TF-IDF remains the algorithm that produces higher quality communities, while *k*-MEANS-CLIQUE-TF-IDF generally exhibits lower performances, even if higher than the other two algorithms (see Figure 2.4(e)-(g)). Furthermore, in terms of efficiency, the gap between the running time of *k*-AGGL-CLIQUE-TF-IDF and both METIS algorithms raises, as depicted in Figure 2.4(h).

Varying the number of vertices. We now evaluate the algorithms on all the evaluation metrics when the number of vertices n in the network varies, while fixing α and β to $1/10$ and 40 respectively. Figure 2.5(a)-(d) shows results when $\mu = 0.1$. As shown, both *k*-AGGL-CLIQUE-TF-IDF and *k*-MEANS-CLIQUE-TF-IDF exhibit better results in terms of quality, whereas both METIS-M and METIS-R generally struggle to discover qualitative communities as n increases (see Figure 2.5(a)-(c)). Moreover, *k*-AGGL-CLIQUE-TF-IDF appears to be independent of n in terms of both modularity and NMI.

Figure 2.5(e)-(h) show the analysis when μ increases to 0.3 . It can be seen

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

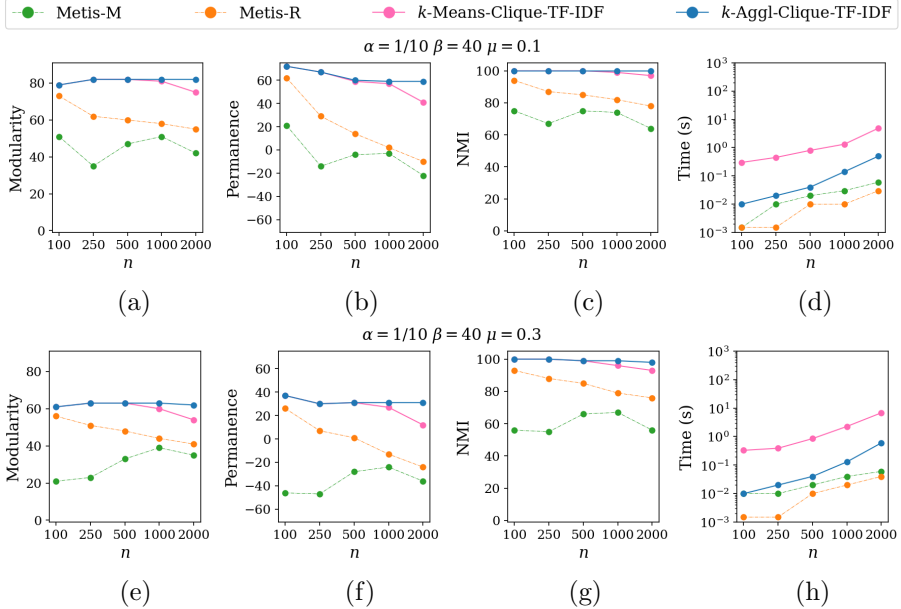


Figure 2.5: Evaluation metrics on synthetic networks when $\alpha = 1/10$ and $\beta = 40$, varying n , for $\mu \in \{0.1, 0.3\}$ (rows).

that *k*-AGGL-CLIQUE-TF-IDF is still the best algorithm in terms of modularity and permanence (see Figure 2.5(e)-(f)). Moreover, even with a higher value of the mixing parameter, our approach is able to discover communities that closely resemble those of the ground truth (see Figure 2.5(g)).

Once again, the main limit of our approach is represented by a higher computation time and a stronger dependency from the number of vertices in the network, as it can be seen in Figure 2.5(d) and in Figure 2.5(h).

Varying maximum vertex degree. Finally, we report the evaluation with respect to different values of the maximum degree parameter α . We fixed $n = 1000$, $\beta = 40$ and we considered $\mu = 0.1$ (Figure 2.6(a)-(d)) and $\mu = 0.3$ (Figure 2.6(e)-(h)).

In Figure 2.6(a)-(b) and Figure 2.6(e)-(f) it can be seen that both *k*-AGGL-CLIQUE-TF-IDF and *k*-MEANS-CLIQUE-TF-IDF are more effective with respect to METIS algorithms in terms of modularity and permanence, with *k*-AGGL-CLIQUE-TF-IDF that appears to be slightly better with respect to

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

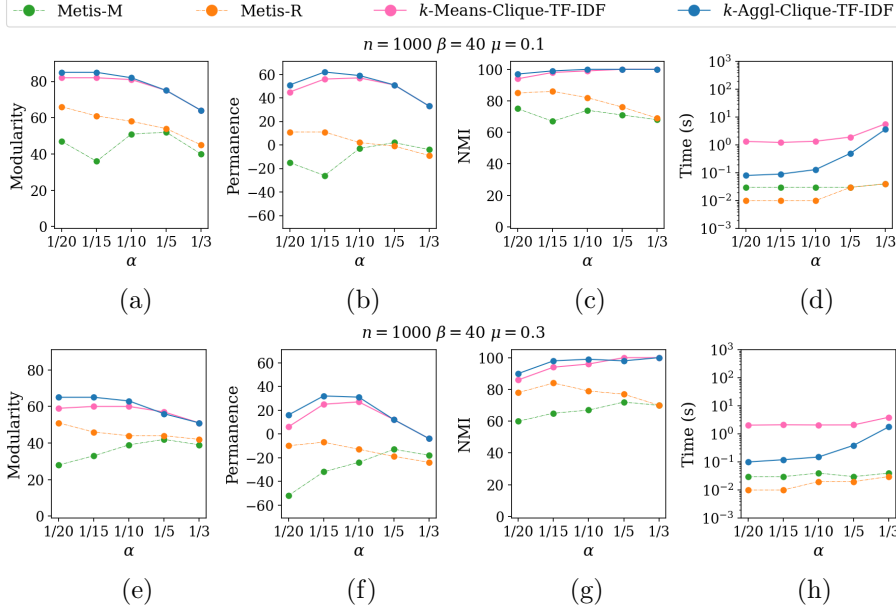


Figure 2.6: Evaluation metrics on synthetic networks when $n = 1000$ and $\beta = 40$, varying α , for $\mu \in \{0.1, 0.3\}$ (rows).

k -MEANS-CLIQUE-TF-IDF as α decreases. The partitioning that is returned by our approach is in general very close to the ground truth, while both METIS-M and METIS-R identify a partitioning that is far from it (see Figure 2.6(c) and Figure 2.6(g)).

Moreover, Figure 2.6(d) and Figure 2.6(h) show that while k -MEANS-CLIQUE-TF-IDF is the least efficient, k -AGGL-CLIQUE-TF-IDF exhibits a stronger dependency of running time as α increases that is primarily attributed to the increased number of maximal cliques and their size. The distribution of the number of maximal cliques with respect to their size for different values of α is depicted in Figure 2.7. In particular, it can be seen that as α increases both the number of cliques and the size of the largest maximal clique increase.

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

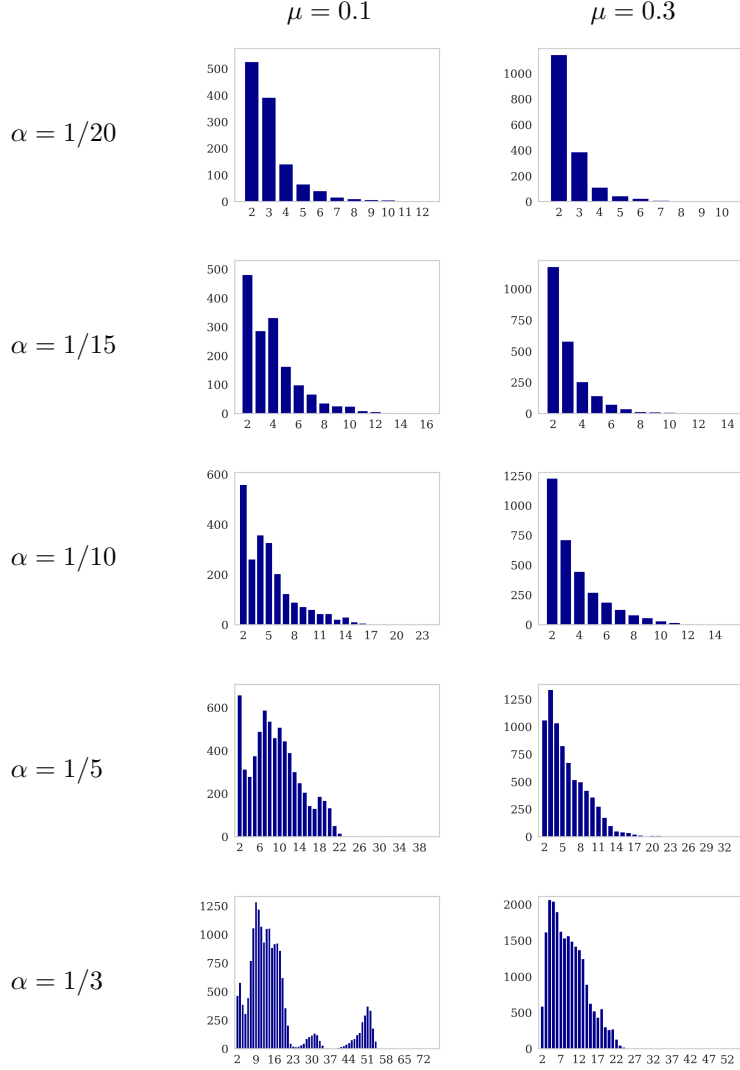


Figure 2.7: Distribution of the number of maximal cliques (y-axis) with respect to their size (x-axis) when $\beta = 40$ and $n = 1000$, for $\alpha \in \{\frac{1}{20}, \frac{1}{15}, \frac{1}{10}, \frac{1}{5}, \frac{1}{3}\}$ (rows) and for $\mu \in \{0.1, 0.3\}$ (columns).

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

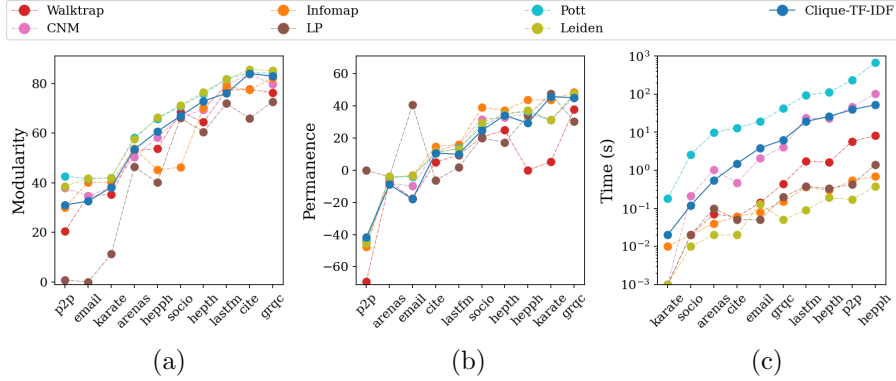


Figure 2.8: Effectiveness and efficiency measures achieved by each algorithm in terms of modularity (a), permanence (b), and running time in seconds (c). The networks on the x-axis are ordered based on the average modularity, on the average permanence, and on CLIQUE-TF-IDF running time, respectively.

2.6 Evaluation of Clique-TF-IDF with Unknown Number of Blocks

In this section we describe an experimental study to evaluate the effectiveness and efficiency of algorithm CLIQUE-TF-IDF, when the number of blocks in the partition is not known a priori.

2.6.1 Experiments on Real-world Networks

We evaluate the efficiency and effectiveness of CLIQUE-TF-IDF in comparison to the algorithms outlined in Section 2.4.3. While CLIQUE-TF-IDF is one of the algorithms that generates partitions with the highest modularity, Leiden consistently outperforms all others in terms of effectiveness (see Figure 2.8(a)). Although Pott also achieves strong modularity results, it does so with significantly higher running times, at least an order of magnitude longer than its competitors. Unlike its effectiveness on synthetic graphs (see Section 2.6.2), CLIQUE-TF-IDF does not produce partitions with high permanence on real-world graphs compared to the other approaches (see Figure 2.8(b)).

Figure 2.8(c) illustrates that although CLIQUE-TF-IDF algorithm has relatively long computation times, they remain lower than those of the Pott algorithm. The large number of maximal cliques in certain networks and the

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

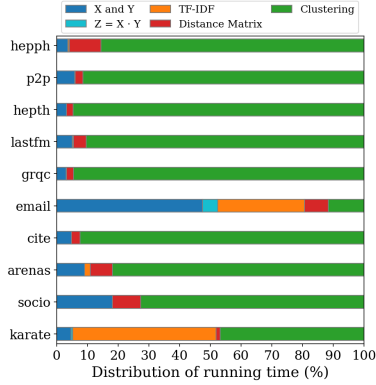


Figure 2.9: Distribution of the running time across each phase of CLIQUE-TF-IDF.

binary search for the optimal number k of blocks are the primary causes of the high computation times. The overall runtime is dominated by the clustering phase, as revealed by a profiling analysis of CLIQUE-TF-IDF (see Figure 2.9). However, two networks stand out as exceptions to this trend, **sociopatterns** and, in particular, **email**. To explain this behavior, we examined the distribution of maximal cliques by size. Figure 2.10 reveals that these two networks have a large number of maximal cliques with a distribution that deviates from the typical scale-free pattern, resembling a binomial distribution in the case of **email**. As a result, a significant portion of time is spent on the computation of the X and Y matrices (blue rectangles in Figure 2.9).

2.6.2 Experiments on Synthetic Networks

Here we discuss the evaluation of CLIQUE-TF-IDF on synthetic graphs produced with LFR compared with other algorithms listed in Section 2.4.3. We analyze how modularity, permanence, NMI, and efficiency are affected by different values of the considered parameters.

Varying community interconnectedness (mixing parameter μ). We fixed $\alpha = 1/10$ and $\beta = 40$, varying μ linearly in the interval $[0.1, 0.5]$ for different values of n to understand how the amount of noise in the network influences the ability of the algorithms to identify communities. As it can be seen in Figure 2.11(a)-(d) when $n = 500$, CLIQUE-TF-IDF is able to find

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

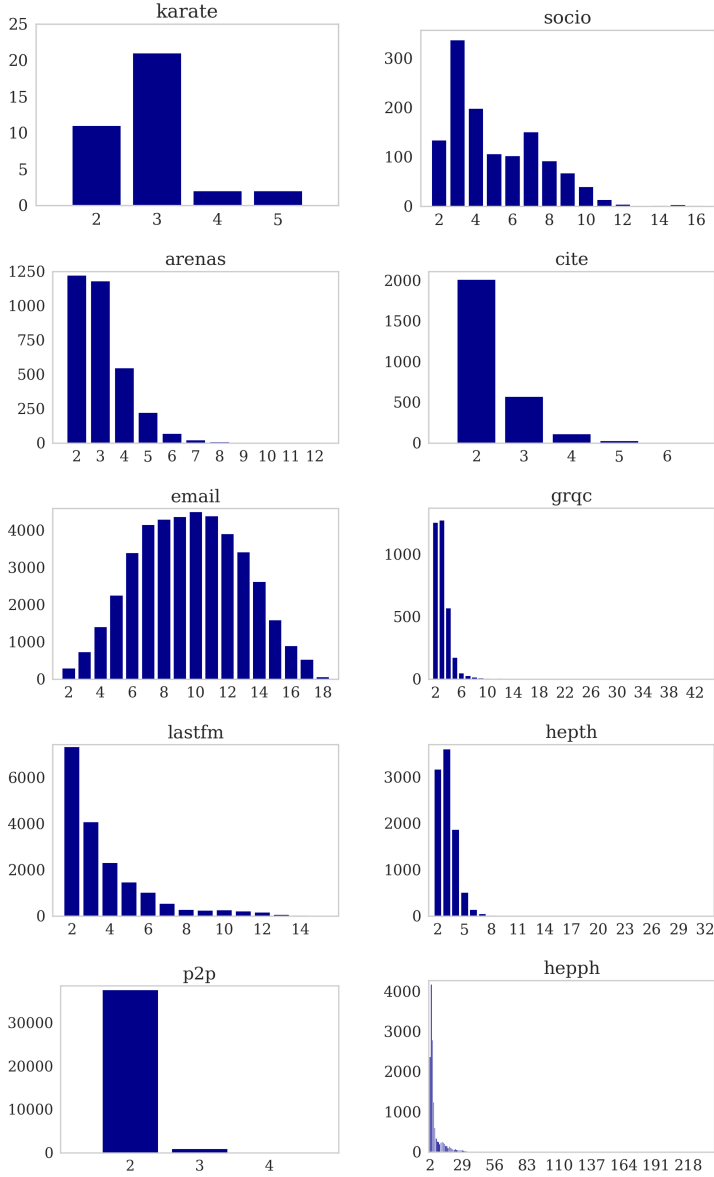


Figure 2.10: Size distribution of maximal cliques for the real-world networks listed in [Table 2.1](#).

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

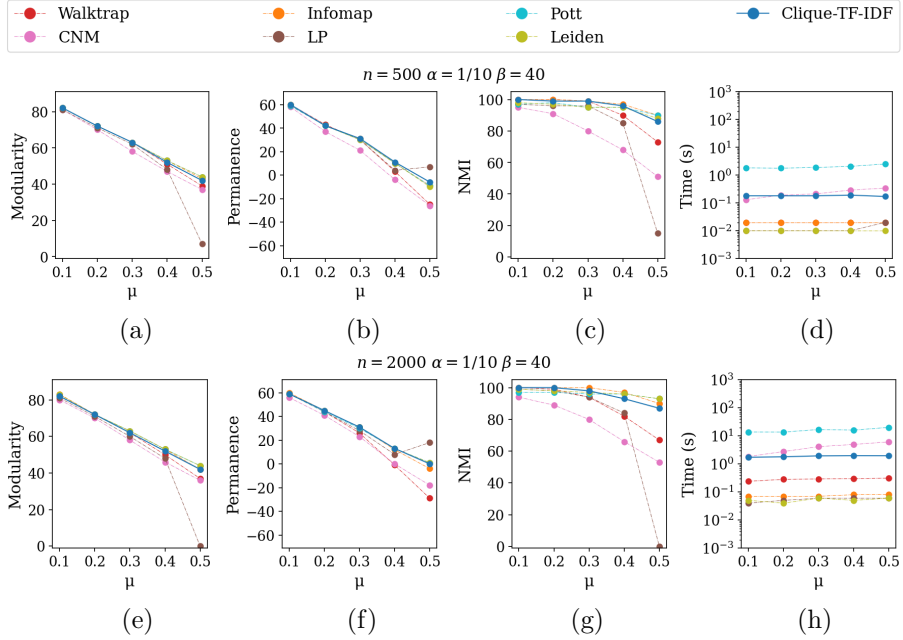


Figure 2.11: Evaluation metrics on synthetic networks when $\alpha = 1/10$ and $\beta = 40$, varying μ , for $n \in \{500, 2000\}$ (rows).

blocks that are qualitative in terms of both modularity and permanence (see Figure 2.11(a)-(b)). More specifically, the modularity and permanence values of our approach are among the best for $\mu \leq 0.3$, while they are comparable for higher values of μ .

When comparing in terms of NMI with the ground truth offered by LFR (Figure 2.11(c)), CLIQUE-TF-IDF is able to maintain high NMI values also when the mixing parameter μ increases, while several algorithms like Walktrap, CNM, and LP decrease drastically their values of NMI when $\mu > 0.3$. In terms of efficiency, our approach has among the largest computation times, that are however comparable with those of CNM and one order of magnitude smaller than those of algorithm Pott (see Figure 2.11(d)). Furthermore, from Figure 2.11(d) it can be observed that CLIQUE-TF-IDF exhibits an independence of its running times with respect to the mixing parameter, while the running times of some other algorithms like CNM slightly increase with μ . Figure 2.11(e)-(h) show an

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

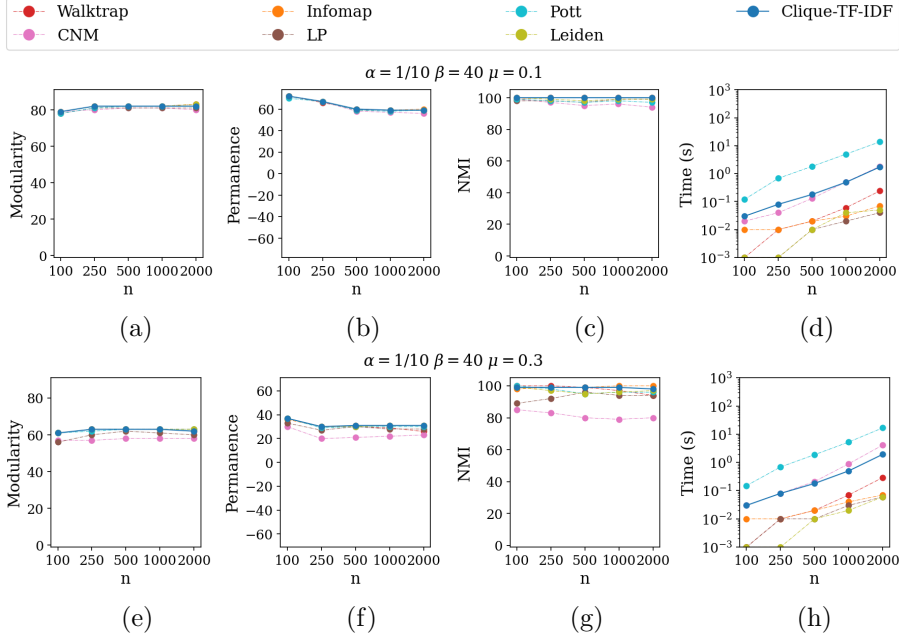


Figure 2.12: Evaluation metrics on synthetic networks when $\alpha = 1/10$ and $\beta = 40$, varying n , for $\mu \in \{0.1, 0.3\}$ (rows).

analogous analysis with $n = 2000$, confirming that the above described behavior of the algorithms is substantially independent of the value of n when the other parameters are fixed.

Varying the number of vertices. In the following, we analyze more in detail how the value of n affects the four evaluation metrics. We fix α and β respectively to $1/10$ and 40 , varying $n \in \{100, 250, 500, 1000, 2000\}$. Figure 2.12 shows the analysis with the mixing parameter $\mu \in \{0.1, 0.3\}$. In particular, it can be seen that CLIQUE-TF-IDF has values of modularity, permanence and NMI that are substantially independent of n , with curves that are always higher or comparable with the best algorithms in the state-of-the-art (see Figure 2.12(a)-(c) and Figure 2.12(e)-(g)). In terms of efficiency, our approach has the same dependency on n of algorithm CNM (Figure 2.12(d) and Figure 2.12(h)). As in the case of real-world networks, algorithm Pott is the most inefficient.

Varying maximum vertex degree. We fixed $n = 1000$ and $\beta = 40$ and

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

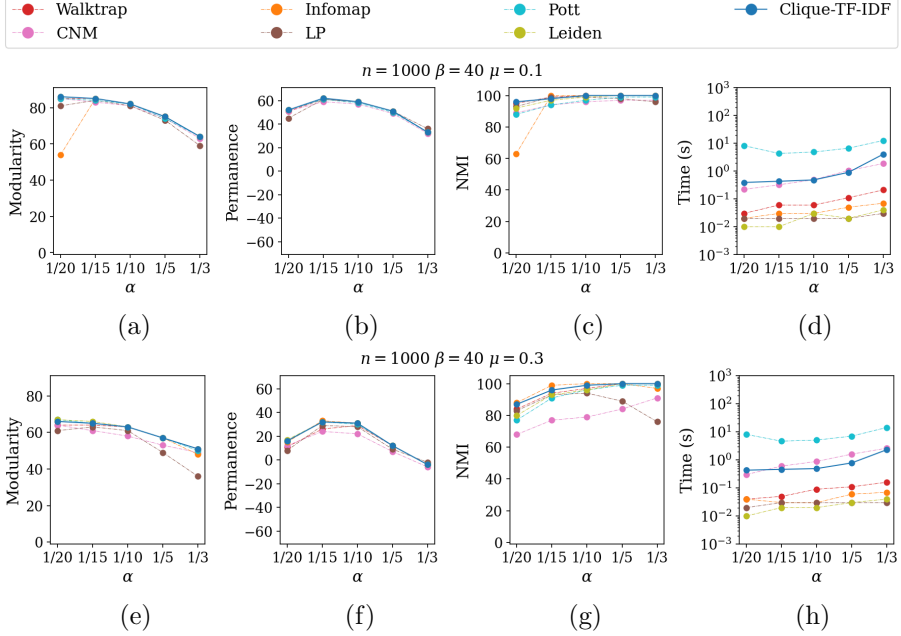


Figure 2.13: Evaluation metrics on synthetic networks when $n = 1000$ and $\beta = 40$, varying α , for $\mu \in \{0.1, 0.3\}$ (rows).

evaluated how the performances of the algorithms vary when α , i.e., the scale-freeness of the network, increases.

When $\mu = 0.1$, in [Figure 2.13\(a\)-\(c\)](#) it can be seen that CLIQUE-TF-IDF can recover clusters that have in general the highest modularity and permanence. Additionally, our approach leads to a value of NMI that is higher and close to the maximum as α increases (i.e., the scale-freeness of the network increases), ensuring results among the best even for lower values of α (see [Figure 2.13\(a\)-\(c\)](#)).

With regard to efficiency, [Figure 2.13\(d\)](#) shows that the running time of most algorithms increases as α grows. Notably, CLIQUE-TF-IDF exhibits a stronger dependency, which can be attributed to the larger number of maximal cliques as it can be seen in [Figure 2.7](#).

In [Figure 2.13\(e\)-\(h\)](#) it can be seen that our approach performs well in all quality metrics even when the noise increases, i.e., when $\mu = 0.3$. In this case,

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

when $\alpha > 1/10$, algorithm LP obtains lower values of modularity and NMI, while CNM has curves of modularity and NMI that are always below those of CLIQUE-TF-IDF. Although we don't include here the graphs, similar results are also obtained with higher values of the mixing parameter μ .

2.7 Experimental Validation of the Clique-based Approach

The experiments discussed in this section aim at showing that the clique-based vertex embedding used in the pipeline of CLIQUE-TF-IDF is a good starting point for computations involving communities in the graph. To achieve this, we applied our clustering phase to two popular graph embedding algorithms, Node2Vec [66] and DeepWalk [67]. Both algorithms produce positional vertex embeddings, where vertex similarity is based on adjacency in the input graph, making them suitable as a preliminary step for positional clustering, which is the focus of this chapter. We used the default values of DeepWalk for both the algorithms. This setting is denoted as D in Figures 2.14 to 2.17. Then, as these algorithms require many parameters in input, we considered different combinations of them. The number of random walks and epochs was fixed at 10, while we varied the embedding dimensionality d within the set $d \in \{32, 64, 128\}$ and the random walk length ℓ within the set $\ell \in \{20, 40, 80\}$. All other parameters were left at their default values. In Figures 2.14 to 2.17, the settings labeled as S, M and L correspond to $d = 32$ and $\ell = 20$, $d = 64$ and $\ell = 40$, and $d = 128$ and $\ell = 80$, respectively.

2.7.1 Real-world Networks

Since both Node2Vec and DeepWalk usually require substantial computation time, even on mid-sized networks, we considered only networks for which these algorithms can compute an embedding within a threshold time of 15 minutes. The results of the embedding comparison, obtained using various configurations of Node2Vec and DeepWalk as discussed above, are shown in Figure 2.14. In terms of effectiveness, Figure 2.14(a) shows that the modularity of the partitions produced by CLIQUE-TF-IDF is larger than that of DeepWalk in all of its settings, while it is comparable than that of Node2Vec (with the only exception of the graph `email`). Similar results can be observed in Figure 2.14(b) with respect to the permanence. Furthermore, variations in the parameters do not

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

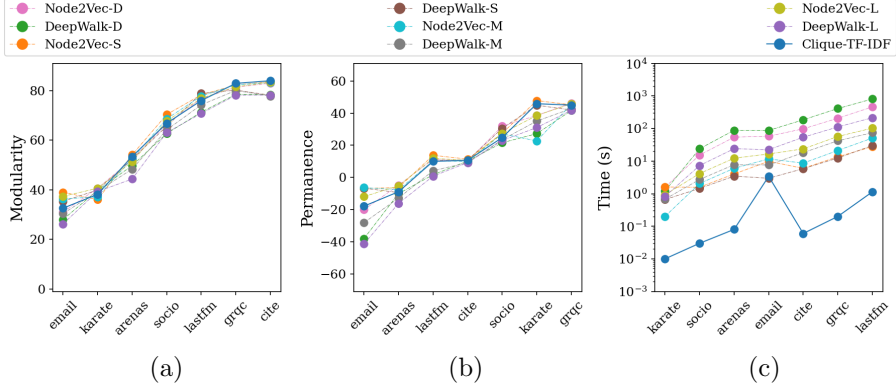


Figure 2.14: Effectiveness and efficiency measures achieved by DeepWalk, Node2Vec and CLIQUE-TF-IDF in terms of modularity (a), permanence (b), and running time in seconds (c). The networks on the x-axis are ordered based on the average modularity, on the average permanence, and on CLIQUE-TF-IDF running time, respectively.

appear to significantly affect the quality of the partitions produced by the algorithms.

Moreover, as depicted in Figure 2.14(a)-(b), it is evident how there is no a clear winner across different settings of Node2Vec and DeepWalk, since the best choice of parameters may depend on the characteristics of the network. In terms of efficiency, Figure 2.14(c) shows that CLIQUE-TF-IDF is significantly faster, by several orders of magnitude, than both DeepWalk and Node2Vec across all their settings (note that the y-axis of Figure 2.14(c) has a logarithmic scale). The increased computation time observed on `email` is due to the huge number of maximal clique in relation to the number of vertices.

2.7.2 Synthetic Networks

Here we discuss the evaluation on synthetic graphs produced with LFR of CLIQUE-TF-IDF compared with the four settings of Node2Vec and DeepWalk. We repeat the same analysis of Section 2.6.2 on all the evaluation metrics.

Varying community interconnectedness (mixing parameter μ). We first fixed $\alpha = 1/10$ and $\beta = 40$, varying μ linearly in the interval $[0.1, 0.5]$ for different values of n , to understand how the amount of noise in the network

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

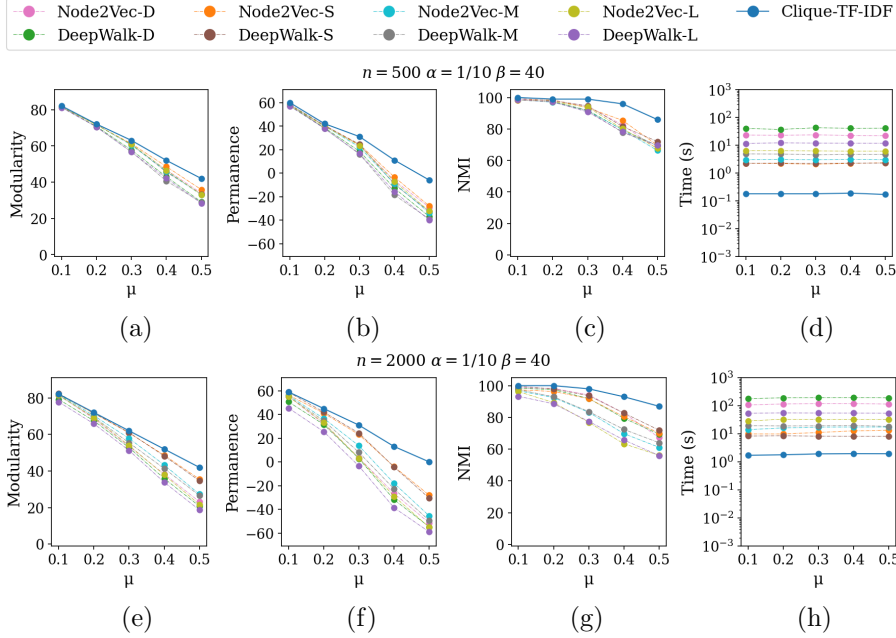


Figure 2.15: Evaluation metrics on synthetic networks when $n = 500$, $\alpha = 1/10$ and $\beta = 40$, varying μ linearly in the interval $[0.1, 0.5]$.

influences the ability of the algorithms to discover strong communities. As it can be seen in Figure 2.15 with $n = 500$, CLIQUE-TF-IDF is able to find more qualitative blocks with respect to Node2Vec and DeepWalk in terms of both modularity and permanence (see Figure 2.15(a)-(b)). More specifically, the curves for modularity and permanence of Node2Vec and Deepwalk start to diverge from our curve when $\mu = 0.3$, showing a clear sensitivity of the embedding quality produced by other algorithms as the noise value increases.

When comparing with the ground truth produced by LFR in terms of NMI (Figure 2.15(c)), CLIQUE-TF-IDF is able to discover most of it also when the mixing parameter μ grows, while other algorithms make more errors in classifying communities. Our approach also outperforms the competitors in terms of efficiency, since we are able to produce an embedding in a time that is several order of magnitude lesser than those of Node2Vec and DeepWalk (see Figure 2.15(d)). It can be observed as well that CLIQUE-TF-IDF is substantially

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

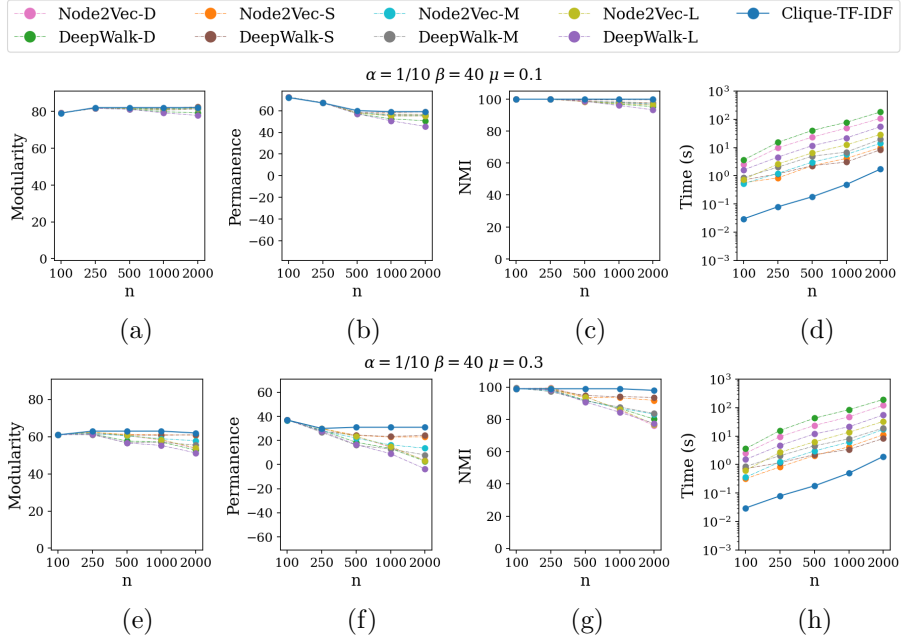


Figure 2.16: Evaluation metrics on synthetic networks when $\alpha = 1/10$ and $\beta = 40$, varying n , for $\mu \in \{0.1, 0.3\}$ (rows).

independent of the noise.

The same analysis is repeated when $n = 2000$. In Figure 2.15(a)-(c), it can be seen that the curves for CLIQUE-TF-IDF in terms of modularity, permanence and NMI are similar, so our approach is independent to the value of n when other parameters are fixed. Instead, performance of other approaches tend to decrease when n grows and this is a further motivation that parameters of these embedding algorithms have a strong dependency on the features of the network.

Varying the number of vertices. Figure 2.16 shows the evaluation with respect to different values of n , choosing $\alpha = 1/10$ and $\beta = 40$. When $\mu = 0.1$, our approach achieves the highest results in terms of quality, and in particular, CLIQUE-TF-IDF obtains values of modularity that are independent of the value of n , while all considered variants of Node2Vec and DeepWalk tend to decrease their quality as n increases (see Figure 2.16(a)-(c)). This phenomenon is amplified when the network is noisier, (i.e., when the mixing parameter

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

increases), with Node2Vec and DeepWalk obtaining results far below those obtained by our algorithm, as it can be seen in Figure 2.16(e)-(g).

Furthermore, when analyzing the running time, Figure 2.16(d) and Figure 2.16(h) show that the embedding phase of CLIQUE-TF-IDF is at least one order of magnitude faster with respect to all the other embedding algorithms considered.

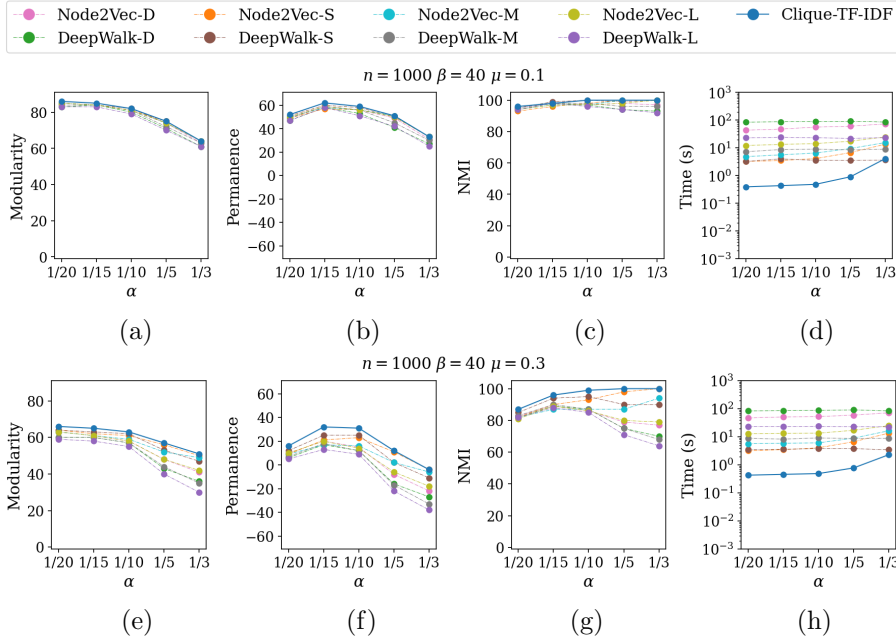


Figure 2.17: Evaluation metrics on synthetic networks when $n = 1000$ and $\beta = 40$, varying α , for $\mu \in \{0.1, 0.3\}$ (rows).

Varying maximum vertex degree. In the end, we analyze how the scale-freeness of the network affects the embedding. We choose n and β to be 1000 and 40 respectively. When $\mu = 0.1$, CLIQUE-TF-IDF obtains values of modularity and permanence that decrease as α increases, but having results that are always better than other approaches (see Figure 2.17(a)-(b)). This reduction is driven by the structure of the network, as discussed in Section 2.4.2.

Furthermore, when comparing to the underlying ground truth, our approach is able to discover most of the communities when $\alpha < 1/10$, while for higher

2. HIGH-MODULARITY GRAPH PARTITIONING THROUGH NLP TECHNIQUES AND MAXIMAL CLIQUE ENUMERATION

values we are able to find the exact ground truth structure when on the contrary Node2Vec and DeepWalk obtain lower results, as it can be seen in [Figure 2.17\(c\)](#).

Once again, our algorithm performs better in terms of modularity, permanence and NMI for all the values of α when the mixing parameter μ increases to 0.3, while other approaches lead to results that diverge from our curves as α increases (see [Figure 2.17\(e\)-\(g\)](#)).

When considering efficiency, we observe in [Figure 2.17\(d\)](#) and [Figure 2.17\(h\)](#) that CLIQUE-TF-IDF exhibits a stronger dependency on the running time with respect to the value of α due to the larger number of maximal cliques (see [Figure 2.7](#)). However, it consistently calculates embeddings more efficiently than Node2Vec and DeepWalk.

Maximal Cliques Summarization: Principles, Problem Classification, and Algorithmic Approaches

3.1 Introduction

In the vast landscape of computer science, few challenges stand as prominently at the crossroads of theory and application as the enumeration of the communities in large graphs. The strictest definition of community is that of a maximal clique. For example, [Figure 3.1](#) shows a graph and its maximal cliques. The two cliques of size 5 in [Figure 3.1b](#) are also maximum cliques. Finding cliques and maximal cliques in graphs has applications in several domains, including web network

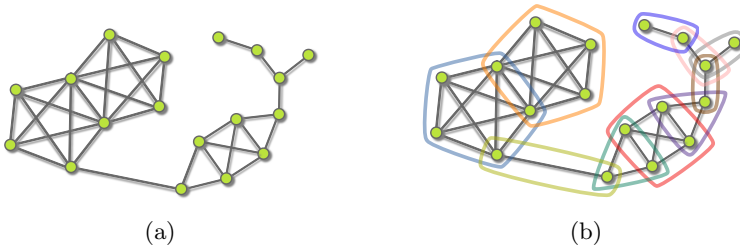


Figure 3.1: (a) An example graph. (b) The maximal cliques of the graph in (a).

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

analysis [76, 77], community detection in real-world networks [41], economics and computational finance [78, 79], biochemistry and genetics [80, 81, 82], microbiology [83], and telecommunications [84, 85, 86].

However, when dealing with large real-world graphs, the huge amount of such maximal cliques makes it unfeasible to use them in several application contexts. For instance, it is not unusual that a graph with less than 10^6 vertices has more than 10^7 maximal cliques [87]. Moreover, maximal cliques have a significant amount of redundancy, as most of them overlap. The same issues occur with some popular relaxations of cliques such as k -cliques [88, 89, 90], k -plexes [33, 34, 35, 36], γ -quasi cliques [38], k -diamonds [91], l -clubs [92], and s -defective cliques [37] (see [39] for a survey).

Chapter contributions. In this chapter¹, we consider the following question (see also [Question 0.2](#)): Given a graph G , can we produce a meaningful summary that represents all the maximal cliques of G in a way that their abundance and redundancy is reduced with respect to certain properties? This is a broad question, and there is no universal method for selecting a subset of relevant maximal cliques that, in some sense, represent all the excluded ones.

We aim at systematically defining various summarization principles, introducing a novel taxonomy, to produce summaries that exhibit different properties and that capture distinct perspectives on the maximal cliques of the graph. The concept of summarizing maximal cliques first appears in [87], where the focus is on finding a subset of maximal cliques that have a significant overlap with those that are excluded. As we show in [Section 3.3](#), this is just one of several possible summarization principles that can be applied when dealing with maximal cliques. To the best of our knowledge, this is the first time that seemingly different research contributions related to maximal cliques summarization have been considered together as different viewpoints on the same goal, providing a classification of all such diverse approaches to maximal clique summarization.

This chapter is devoted both to classifying existing problems in the literature within a comprehensive taxonomy and to introducing new research lines that naturally arise when summarization principles are combined with traditional problem targets. Indeed, we also classify summarization problems into four families, based on whether the summary is produced by considering Boolean properties or quality measures. This chapter is primarily focused on the algorithmic aspects of the problems discussed, with a detailed exploration of methods and techniques presented in the literature.

¹The content of this chapter appeared in [93] and is joint work with Irene Finocchi and Maurizio Patrignani.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

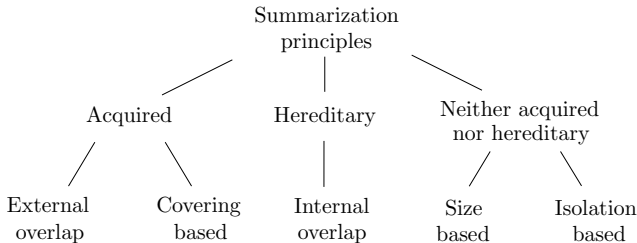


Figure 3.2: A taxonomy of maximal cliques summarization principles.

Chapter organization. The rest of the chapter is organized as follows. In [Section 3.2](#), we provide the needed definitions and preliminaries; in [Section 3.3](#), we introduce a taxonomy of maximal clique summarization principles; in [Section 3.4](#), we introduce a taxonomy of summarization problems and we review both those problems that have been studied in the literature and new problems that may naturally arise; in [Section 3.5](#), we discuss problems that involve variations in the input graph or in the expected output; in [Section 3.6](#), we outline promising research directions using the classification introduced in [Section 3.4](#); finally, in [Section 3.7](#), we review the algorithmic contributions found in the literature addressing the problems listed in [Section 3.4](#).

3.2 Preliminaries

We refer to [Chapter 1](#) for basic definitions on graphs. Let $G = (V, E)$ be a simple undirected graph, and let n denote the number of vertices of G . Recall that, for a vertex $v \in V$, $\delta(v)$ denotes its degree and $N(v)$ its neighborhood, and that for any vertex subset $X \subseteq V$, $G[X]$ denotes the subgraph of G induced by X .

3.3 Summarization Principles

Several principles can be adopted to pinpoint the relevant information provided by the maximal cliques of a graph, without necessarily relying on their exhaustive enumeration. The choice of the simplification principles depends on the application context at hand. In this section we discuss the main summarization approaches adopted so far in the literature, in the context of a new taxonomical framework (see [Figure 3.2](#)).

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Let $S \subseteq \mathcal{M}(G)$ be any subset of maximal cliques of an undirected graph G . We refer to S as a *summary* of $\mathcal{M}(G)$. In particular, if $|S| = s$, we call S an *s-summary*. Summarization principles can be classified as acquired or hereditary:

- A summarization principle P is *acquired* if for any summary $S \subseteq \mathcal{M}(G)$ that satisfies P , it holds that every S' such that $S \subseteq S' \subseteq \mathcal{M}(G)$ also satisfies P . For an acquired principle P it makes sense to search for a summary of reduced size, since the very $\mathcal{M}(G)$ would be a summary satisfying P .
- A summarization principle P is *hereditary* if for any summary $S \subseteq \mathcal{M}(G)$ that satisfies P , it holds that every $S'' \subseteq S$ also satisfies P . Observe that a hereditary principle P is always satisfied by the empty summary. Hence, such a principle is usually used in combination with other (non-hereditary) principles.

Acquired and hereditary principles can be further distinguished. Each of the remaining subsections focuses on one of these detailed principles, which in turn corresponds to one of the leaves of the taxonomy tree of [Figure 3.2](#).

3.3.1 Size-based Summarization

Size-based criteria may be helpful in applications where large maximal cliques are more relevant than smaller ones. For instance, one could list all maximal cliques with a given size k , or only those whose size is at least k . Another size-based strategy could be to find the largest (i.e., maximum) clique or to list all the maximum cliques. From the examples provided above it is apparent that this principle is neither acquired nor hereditary.

When a size-based criterion is adopted, the degree of relevance of a maximal clique depends exclusively on its own properties (i.e., the number of its vertices), and not on its relationship with other cliques. This is different from other summarization principles discussed in the following.

Practical applications of size-based summarization approaches include filtering the graph by removing noisy or peripheral regions and presenting only its denser portion to the user. In other cases, the goal is to produce a succinct core that highlights the most connected nodes, implicitly providing a clear description of their connectivity patterns.

Size-based approaches might be not always satisfactory, especially in applications where the summary should convey information about the entire graph.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

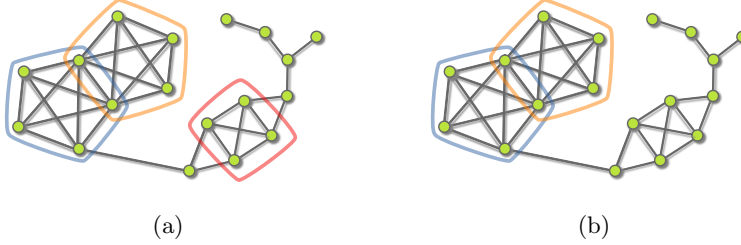


Figure 3.3: (a) The maximal cliques in a size-based summary with $k = 4$. (b) The maximal cliques in a size-based summary with $k = 5$.

For instance, when listing maximal cliques of size at least k , some areas of the graph may not be properly summarized. For example, Figure 3.3b shows an example of a size-based summary S when $k = 5$: in this summary all the rightmost portion of the graph is neglected. The summary S in Figure 3.3a is instead obtained when $k = 4$. Notice that the summary for the larger value of k (Figure 3.3b) is a subset of the summary for the smaller value of k (Figure 3.3a). Also, it is often the case that large cliques are strongly overlapped, yielding a representation that may be redundant for the denser portions of the graph and incomplete for the sparser portions of it.

3.3.2 Overlap-based Summarization

The enumeration of maximal cliques often produces cliques that are strongly overlapped. In order to grasp the extent of the overlap phenomenon, consider the following example. Let C be a maximal clique of size n as shown in Figure 3.4a. Let $(v_1, v'_1), (v_2, v'_2), (v_3, v'_3), \dots$ be a sequence of mutually non adjacent edges of C and let $C_1 = C \setminus \{v_1, v'_1\}$, $C_2 = C \setminus \{v_1, v'_1, v_2, v'_2\}$, $C_3 = C \setminus \{v_1, v'_1, v_2, v'_2, v_3, v'_3\}$, $\dots$. By removing edge (v_1, v'_1) from C , we obtain two maximal cliques of size $n - 1$, namely, $C_1 \cup \{v_1\}$ and $C_1 \cup \{v'_1\}$ (see Figure 3.4b). By removing the second edge (v_2, v'_2) , we obtain four maximal cliques of size $n - 2$, namely, $C_2 \cup \{v_1, v_2\}$, $C_2 \cup \{v_1, v'_2\}$, $C_2 \cup \{v'_1, v_2\}$, and $C_2 \cup \{v'_1, v'_2\}$ (see Figure 3.4c). When we remove the i -th edge, we obtain 2^i maximal cliques of size $n - i$ that share clique C_i of size $n - 2i$ (see Figure 3.4d for $i = 3$). In conclusion, the elision of edges from the original maximal clique produces an exponential number of maximal cliques of linearly smaller size. All of them overlap in a portion that is also of linear size.

Since it is common in different domains that some edges are missing, due to

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

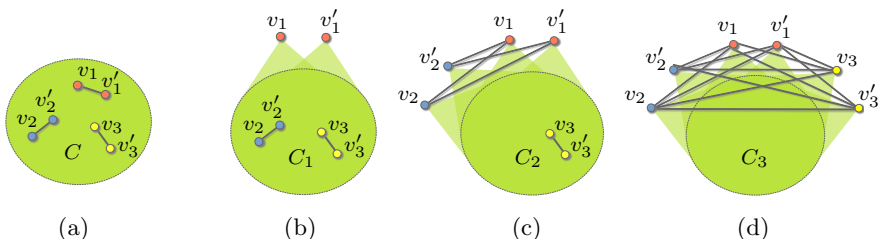


Figure 3.4: (a) A clique C with three independent edges highlighted. (b), (c), (d) The effect of the removal of the edges.

elision, noise or uncertainty in the available data, the presence of a high number of strongly overlapped maximal cliques is very common. Therefore, in several applications a single maximal clique may account for all the other maximal cliques that are strongly overlapped with it. In these cases, a possible strategy to summarize the maximal cliques is that of avoiding overlapped ones. This could lead to two different objectives: (i) *external-overlap summarization* requires that the maximal cliques in $\mathcal{M}(G)$ not contained in summary S have a strong overlap with at least one of those contained in S ; and (ii) *internal-overlap summarization* requires that the maximal cliques contained in S have a small overlap among themselves. Both these objectives motivate the quest for a precise formal definition of when a maximal clique C is considered ‘represented’ or ‘visible’ by another maximal clique C' . The *local visibility* [87] of a maximal clique C with respect to another maximal clique C' is defined as $\mathcal{V}_{C'}(C) = \frac{|C \cap C'|}{|C|}$. Notice that local visibility is neither transitive nor symmetric. In the following sections we describe how local visibility can be used both for external-overlap and internal-overlap summarization.

External-overlap summarization. In this kind of summarization the aim is ensuring that the maximal cliques not contained in S are somehow represented by those in S . For example, one could exclude from S those maximal cliques that are strongly overlapped with some maximal clique of S or search for a summary S that maximizes a measure of the overlap with the maximal cliques that are not in S . Observe that external-overlap summarization is an acquired principle.

Following the notation used in [87], the *visibility* of a maximal clique $C \in$

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

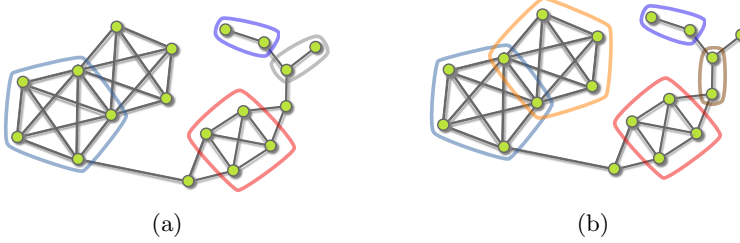


Figure 3.5: (a) A τ -visible summary S with $\tau = 0.4$. Each excluded maximal clique has at least 40% of its vertices included into a maximal clique in S . (b) Another τ -visible summary S with $\tau = 0.4$.

$\mathcal{M}(G)$ with respect to S , denoted as $\mathcal{V}_S(C)$, is defined as:

$$\mathcal{V}_S(C) = \max_{C' \in S} \mathcal{V}_{C'}(C) = \max_{C' \in S} \frac{|C \cap C'|}{|C|} \quad (3.1)$$

A summary S of a graph G is τ -visible, with $0 \leq \tau \leq 1$, if the visibility of each maximal clique $C \in \mathcal{M}(G)$ with respect to S is at least τ , i.e., $\mathcal{V}_S(C) \geq \tau$. For example, Figures 3.5a and 3.5b show two possible τ -visible summaries when $\tau = 0.4$. If $\tau = 0$, any $S \subseteq \mathcal{M}(G)$ is τ -visible. If $\tau = 1$, the summary S necessarily coincides with $\mathcal{M}(G)$. Notice that a τ -visible summary S may in general not be a covering of V , i.e., $\exists v \in V \mid v \notin C, \forall C \in S$. The concept of τ -visibility can be used to define either Boolean properties for a summary S (asking, for example, that S is τ -visible) or an external-overlap-based quality measure for S (considering, for example, the maximum τ for which S is τ -visible).

External-overlap summarization is especially helpful in those applications that would need to review the whole set of maximal cliques, but that, for efficiency reasons, restrict their analysis to a subset of them, each representative of a group of “similar” ones.

Internal-overlap summarization. This summarization principle aims at ensuring that the maximal cliques in S have a small overlap and is especially recommended in those applications that need to cluster strongly connected vertices into groups that are “unrelated” among themselves. The overlap among the cliques in S can be measured in terms of the local visibility between each (ordered) pair of them.

We say that the *local independence* $\mathcal{I}(C, C')$ of two cliques C and C' is the maximum local visibility of C with respect to C' and vice-versa. More formally,

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

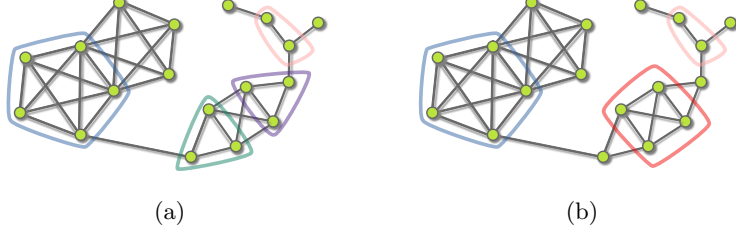


Figure 3.6: Examples of internal overlap summarization. (a) A σ -independent summary S with $\sigma = 0$. (b) A ρ -edge-independent summary S with $\rho = 0$.

$\mathcal{I}(C, C') = \max\{\mathcal{V}_C(C'), \mathcal{V}_{C'}(C)\} = \max\left\{\frac{|C \cap C'|}{|C|}, \frac{|C \cap C'|}{|C'|}\right\} = \frac{|C \cap C'|}{\min(|C|, |C'|)}$. A summary S is σ -independent, with $0 \leq \sigma \leq 1$, if for each pair C and C' of maximal cliques in S we have $\mathcal{I}(C, C') \leq \sigma$. For example, Figure 3.6a shows a σ -independent summary when $\sigma = 0$. If $\sigma = 1$, any $S \subseteq \mathcal{M}(G)$ is σ independent. If $\sigma = 0$, a σ -independent summary S is composed by non-mutual intersecting maximal cliques.

When $\sigma = 0$, a stronger concept of independency can be enforced. Namely, one could ask that two maximal cliques in S are not connected by many edges. The *external edge set* of a maximal clique $C \in \mathcal{M}(G)$, denoted as $\mathcal{E}(C)$, is defined as $\mathcal{E}(C) = \{(u, v) \mid u \in C \text{ XOR } v \in C\}$. Let C and C' be two maximal cliques of size c and c' , respectively, both belonging to a 0-independent summary S . Observe that any vertex $v \in C$ is adjacent to at most $c' - 1$ vertices of C' (otherwise, C' would not be maximal). Therefore, the maximum number of edges connecting C with C' is $e_{\max}(C, C') = \min\{c'(c-1), c(c'-1)\} = c \cdot c' - \max\{c, c'\}$. We define the *local edge independence* $\mathcal{E}(C, C')$ of two maximal cliques C and C' as

$$\mathcal{E}(C, C') = \frac{|\mathcal{E}(C) \cap \mathcal{E}(C')|}{e_{\max}(C, C')} \quad (3.2)$$

A summary S is ρ -edge-independent, with $0 \leq \rho \leq 1$, if S is 0-independent and if $\mathcal{E}(C, C') \leq \rho$ for every pair of maximal cliques C and C' in S . For example, Figure 3.6b shows a ρ -edge-independent summary when $\rho = 0$. If $\rho = 0$, a ρ -edge-independent summary S consists of unconnected maximal cliques. Conversely, if $\rho = 1$, any $S \subseteq \mathcal{M}(G)$ is ρ -edge-independent.

Observe that internal-overlap summarization is a hereditary principle: a summary S remains σ -independent or ρ -edge-independent even when removing a maximal clique from it.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

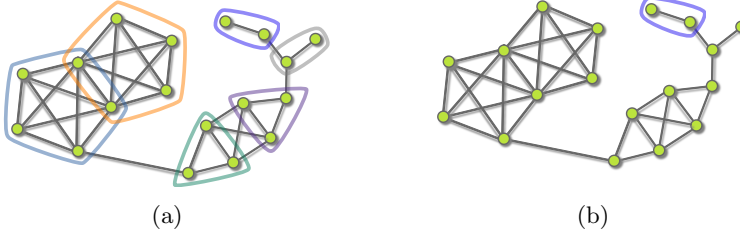


Figure 3.7: (a) A c -covering summary S with $c = 1$: each vertex of the graph is contained into at least one maximal clique in S . (b) A summary S with all the ℓ -isolated maximal cliques when $\ell = 1$: the chosen maximal cliques have less than k outgoing edges.

3.3.3 Covering-based Summarization

Intuitively, when searching for a reduced size summary, maximizing the “coverage” of the graph vertices, i.e., the number of vertices that are contained into at least one maximal clique of the summary, may be a way to reduce the redundancy of the summarization. This principle is sometimes called in the literature “diversification” [94], since it is assumed that the maximal cliques in the summary have a small overlap. More formally, given a set of maximal cliques $S \subseteq \mathcal{M}(G)$ of a graph G , the *coverage* of S , denoted as $c(S)$, is the set of vertices of G covered by the maximal cliques in S , i.e., $c(S) = \bigcup_{C \in S} C$. We say that S is a c -*covering* of G , with $0 \leq c \leq 1$, if $\frac{|c(S)|}{|V(G)|} \geq c$. For example, Figure 3.7a shows a c -covering summary when $c = 1$.

Additionally, the *edge-coverage* of S , denoted as $ec(S)$, is the set of edges of G covered by the maximal cliques in S , i.e., $ec(S) = \bigcup_{C \in S} E(C)$. We say that S is a c -*edge-covering* of G , with $0 \leq c \leq 1$, if $\frac{|ec(S)|}{|E(G)|} \geq c$. The fact that covering-based summarization is an acquired principle justifies the search for a reduced size summary.

Covering-based summarization can be used to produce summaries that represent a large portion of the vertices or edges, ensuring that a wide and meaningful area of the graph is captured. Vertex-covering produces a summary where a large portion of the vertices is represented, while edges could be significantly filtered out. Edge-covering coincides with distributing a large portion of the edges into cliques, actually producing a decomposition of the graph.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

3.3.4 Isolation-based Summarization

For some application contexts, maximal cliques that are poorly connected to the rest of the graph are more relevant than maximal cliques that are somehow intertwined with it. These applications search for bunches of vertices that are “independent”, that is, strongly connected among themselves and weakly connected to the rest of the graph. This kind of summaries are used, for example, to coarsen real-world networks, such as the web graph, at various levels of abstraction while preserving their scale-free properties [95, 96, 97]. The concept of clique independency is captured by the notion of ℓ -isolation [98, 99, 100]. Given a graph $G = (V, E)$ a set of vertices $I \subseteq V$ of size k is *avg- ℓ -isolated* if the number $\delta_e(I)$ of edges that have one endpoint in I and one endpoint in $V \setminus I$ is less than $\ell \cdot k$. A summary S containing all the avg- ℓ -isolated maximal cliques when $\ell = 1$ is shown in Figure 3.7b. A set of vertices $I \subseteq V$ is *max- ℓ -isolated* if each vertex $v \in I$ has less than ℓ neighbors outside I . A set of vertices $I \subseteq V$ is *min- ℓ -isolated* if at least one vertex $v \in I$ has less than ℓ neighbors outside I . Clearly, max- ℓ -isolatedness implies avg- ℓ -isolatedness, which in turn implies min- ℓ -isolatedness, but not vice versa. Number ℓ is sometimes called the *isolation factor*.

This summarization principle asks, for each concept of isolation above and for some value of ℓ , to focus on maximal cliques that are also ℓ -isolated (*ℓ -isolated maximal cliques*). In [99, 100] a second target is also considered: computing ℓ -isolated cliques that are also “maximal”, in the sense that they are not contained in other ℓ -isolated cliques (*maximal ℓ -isolated cliques*). In the remainder we only focus on ℓ -isolated maximal cliques, since the second target is out of the scope of this chapter: an ℓ -isolated clique that is not contained in any other ℓ -isolated clique may not be maximal with respect to the graph. It can be shown, however, that maximal ℓ -isolated cliques are a superset of ℓ -isolated maximal cliques [98]. It follows that one could obtain all ℓ -isolated maximal cliques by filtering from maximal ℓ -isolated cliques those that are not maximal with respect to the graph.

Observe that this summarization principle is neither acquired nor hereditary.

3.4 Problem Classification

For each summarization principle several different problems can be considered. We introduce a classification of such problems into four classes based on the type of property that the desired summary should satisfy (see Figure 3.8). When this property is expressed with respect to each single maximal clique, we have a

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

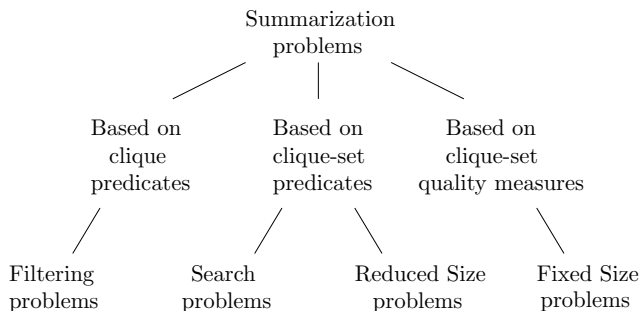


Figure 3.8: A taxonomy of maximal cliques summarization problems.

Filtering problem (see [Section 3.4.1](#)). When, instead, the property is expressed with respect to a whole summary S , then we have either a Search problem (see [Section 3.4.2](#)) or an optimization problem where the summary should have a small size (see [Section 3.4.3](#)). Finally, when the summary size is fixed, we might be interested in finding the best summary with respect to a given quality measure (see [Section 3.4.4](#)).

An overview of the problems discussed or proposed in this section is provided in [Tables 3.1 to 3.3](#), where the problems are grouped according to the four classes introduced in this section. For each problem, the summarization principles are listed, along with whether the problem is open or the references to the literature where it has been studied.

3.4.1 Filtering

In this kind of problems, the summarization principles are used to construct a Boolean property that may or may not be satisfied by any maximal clique in $\mathcal{M}(G)$. The target is that of efficiently producing the summary $S \subseteq \mathcal{M}(G)$ composed by those maximal cliques that satisfy the given property. Observe that S , which could be empty, is unique.

Examples of Filtering problems include the following.

Problem 3.1 (MAXIMAL CLIQUES BY SIZE). Given a graph G and a positive integer s , find all maximal cliques of size at least s .

[Problem 3.1](#) is addressed in [\[101\]](#) and in [Section 3.7.1](#).

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Filtering	Problem 3.1	MAXIMAL CLIQUES BY SIZE <i>Summarization principles: size</i> [101, 102]
	Problem 3.2	ℓ -ISOLATED MAXIMAL CLIQUES <i>Summarization principles: isolation</i> [98, 99, 100]
	Problem 3.3	ℓ -ISOLATED MAXIMAL CLIQUES BY SIZE <i>Summarization principles: size, isolation</i> Open
Search	Problem 3.4	FIND ANY τ -VISIBLE σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY <i>Summarization principles: external overlap, internal overlap</i> Open
	Problem 3.5	FIND ANY c -COVERING σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY <i>Summarization principles: internal overlap, covering</i> Open in general. $c = 1, \sigma = 0$, non-maximal: [103, 104, 105]
	Problem 3.6	FIND ANY c -EDGE-COVERING σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY <i>Summarization principles: internal overlap, covering</i> Open in general. $c = 1, \sigma = 0$: [106, 107] (non-maximal: [108, 109, 110])
	Problem 3.7	c -COVERING ρ -EDGE-INDEPENDENT MAXIMAL CLIQUE SUMMARY <i>Summarization principles: internal overlap, covering</i> Open in general. $\rho = 0$, non-maximal: [111, 112, 113, 114, 115]
	Problem 3.8	c -EDGE-COVERING ρ -EDGE-INDEPENDENT MAXIMAL CLIQUE SUMMARY <i>Summarization principles: internal overlap, covering</i> Open in general. $\rho = 0$, non-maximal: [116, 117, 118, 119, 120, 121]

Table 3.1: An overview of Filtering and Search problems discussed or proposed in Section 3.4.

Problem 3.2 (ℓ -ISOLATED MAXIMAL CLIQUES). Given a graph G and a positive integer ℓ , find all maximal cliques that are ℓ -isolated with respect to an isolation concept (max-, avg-, min- ℓ -isolation).

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Problem 3.2 is addressed in [98, 99, 100], which are surveyed in [Section 3.7.4](#), and also discussed in [Chapter 4](#).

The Boolean property used for filtering can be obtained by combining different summarization principles. One example is the following.

Problem 3.3 (ℓ -ISOLATED MAXIMAL CLIQUES BY SIZE). Given a graph G and two positive integers ℓ and s , find all maximal cliques that are ℓ -isolated with respect to an isolation concept and that have size at least s .

To the best of our knowledge, [Problem 3.3](#) has never been addressed.

3.4.2 Search

In this kind of problems, the summarization principles are used to construct a Boolean property that may or may not be satisfied by a subset S of $\mathcal{M}(G)$. The aim is that of producing any summary $S \subseteq \mathcal{M}(G)$ that satisfies the property, provided it exists. Usually, the Boolean property combines at least two summarization principles, where the first is hereditary and the second is acquired. Examples include the following.

Problem 3.4 (FIND ANY τ -VISIBLE σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY). Given a graph G and two values τ and σ in $[0, 1]$, find a τ -visible summary $S \subseteq \mathcal{M}(G)$ that is also σ -independent.

Problem 3.5 (FIND ANY c -COVERING σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY). Given a graph G and two values c and σ in $[0, 1]$, find a c -covering summary $S \subseteq \mathcal{M}(G)$ that is also σ -independent.

When $c = 1$ and $\sigma = 0$, [Problem 3.5](#) corresponds to partitioning the vertices of G into non-overlapped maximal cliques. When cliques are non-necessarily maximal it has been addressed in [103, 104, 105].

Problem 3.6 (FIND ANY c -EDGE-COVERING σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY). Given a graph G and two values c and σ in $[0, 1]$, find a c -edge-covering summary $S \subseteq \mathcal{M}(G)$ that is also σ -independent.

When $c = 1$ and $\sigma = 0$, [Problem 3.6](#) corresponds to partitioning the edges of G into non-overlapped maximal cliques and has been studied in [106, 107]. When cliques are non-necessarily maximal it has been addressed in [108, 109, 110] and surveyed in [122].

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Problem 3.7 (c -COVERING ρ -EDGE-INDEPENDENT MAXIMAL CLIQUE SUMMARY). Given a graph G and two values c and ρ in $[0, 1]$, find a c -covering summary $S \subseteq \mathcal{M}(G)$ that is also ρ -edge-independent.

When $\rho = 0$, [Problem 3.7](#) has been studied in [[111](#), [112](#), [113](#), [114](#), [115](#)] in a variant where the goal is to maximize c for cliques that are non-necessarily maximal. This variant is known with the name CLUSTER VERTEX DELETION or, equivalently, INDEPENDENT UNION OF CLIQUES.

Problem 3.8 (c -EDGE-COVERING ρ -EDGE-INDEPENDENT MAXIMAL CLIQUE SUMMARY). Given a graph G and two values c and ρ in $[0, 1]$, find a c -edge-covering summary $S \subseteq \mathcal{M}(G)$ that is also ρ -edge-independent.

When $\rho = 0$, [Problem 3.8](#) has been studied in [[116](#), [117](#), [118](#), [119](#), [120](#), [121](#)] in a variant where the goal is to maximize c for cliques that are non-necessarily maximal. This variant is known with the name CLUSTER DELETION.

3.4.3 Reduced Size

In this kind of problems, the summarization principles are used to construct a Boolean property that may or may not be satisfied by any subset S of $\mathcal{M}(G)$. The target is that of efficiently producing a summary $S \subseteq \mathcal{M}(G)$ that satisfies the property and has small size. Usually, the Boolean property refers to an acquired summarization principle, i.e., if S satisfies the property, any superset $S' \supseteq S$, also satisfies it. For example τ -visibility for a specific τ is one such properties. Another example is c -covering for a specific c . We provide hereunder some examples of Reduced Size problems.

Problem 3.9 (τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY). Given a graph G and a value τ , with $0 \leq \tau \leq 1$, find a τ -visible summary $S \subseteq \mathcal{M}(G)$ of small size.

This problem coincides with the enumeration of maximal cliques when $\tau = 1$. [Problem 3.9](#) is addressed in [[87](#), [123](#), [124](#)], which are surveyed in [Section 3.7.2](#).

Problem 3.10 (c -COVERING MAXIMAL CLIQUE SMALL SUMMARY). Given a graph G and a value c , with $0 \leq c \leq 1$, find a c -covering summary $S \subseteq \mathcal{M}(G)$ of small size.

Problem 3.11 (c -EDGE-COVERING MAXIMAL CLIQUE SMALL SUMMARY). Given a graph G and a value c , with $0 \leq c \leq 1$, find a c -edge-covering summary $S \subseteq \mathcal{M}(G)$ of small size.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Reduced Size	Problem 3.9	τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY <i>Summarization principles: external overlap</i> [87, 123, 124]
	Problem 3.10	c -COVERING MAXIMAL CLIQUE SMALL SUMMARY <i>Summarization principles: covering</i> Open
	Problem 3.11	c -EDGE-COVERING MAXIMAL CLIQUE SMALL SUMMARY <i>Summarization principles: covering</i> Open in general. $c = 1$, non-maximal: [108, 109, 125, 126]
	Problem 3.12	τ -VISIBLE c -COVERING MAXIMAL CLIQUE SMALL SUMMARY <i>Summarization principles: external overlap, covering</i> Open
	Problem 3.13	τ -VISIBLE σ -INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY <i>Summarization principles: external overlap, internal overlap</i> Open
	Problem 3.14	c -COVERING σ -INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY <i>Summarization principles: internal overlap, covering</i> Open in general. $c = 1$, $\sigma = 0$, non-maximal: [103, 104, 105]

Table 3.2: An overview of Reduced Size problems discussed or proposed in Section 3.4.

When $c = 1$, **Problem 3.11** has been studied for (non-necessarily maximal) cliques in [108, 109, 125, 126]. The survey [122] refers to this variant.

The considered Boolean property could be obtained by combining different summarization principles. One example is the following.

Problem 3.12 (τ -VISIBLE c -COVERING MAXIMAL CLIQUE SMALL SUMMARY). Given a graph G and two values τ and c in $[0, 1]$, find a τ -visible summary $S \subseteq \mathcal{M}(G)$ of small size that is also c -covering.

Also observe that any problem formulated as “find any” can be translated into a Reduced Size problem, where the target is that of producing a small-size summary $S \subseteq \mathcal{M}(G)$ among those that satisfy the property.

For example compare the following with **Problem 3.4**.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Problem 3.13 (τ -VISIBLE σ -INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY). Given a graph G and two values τ and σ in $[0, 1]$, find a τ -visible summary $S \subseteq \mathcal{M}(G)$ of small size that is also σ -independent.

Compare the following with [Problem 3.5](#).

Problem 3.14 (c -COVERING σ -INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY). Given a graph G and two values c and σ in $[0, 1]$, find a c -covering summary $S \subseteq \mathcal{M}(G)$ of small size that is also σ -independent.

When $c = 1$ and $\sigma = 0$, [Problem 3.14](#) corresponds to partitioning the vertex-set of a graph in such a way that each block of the partition corresponds to a maximal clique. The variant of this problem when cliques are non-necessarily maximal has been studied in [\[103, 104, 105\]](#).

Regarding the objective of reducing the size of S , a common practice in the literature is that of finding summaries whose cardinality is smaller than the cardinality of the smallest summaries computed by previous algorithms on the same datasets. In absolute terms “small size” could be interpreted as “minimum size” yielding more challenging versions of the above problems. A relaxed version of searching for minimum size summaries could be that of searching for summaries that are minimal, that is, they don’t satisfy the property if any maximal clique is removed.

Regarding the efficiency, the proposed algorithms are sometimes compared among themselves, with [Algorithm 1.1](#) as a baseline.

3.4.4 Fixed Size

In this kind of problems the size k of the maximal clique summary is fixed and the summarization principles are used to obtain a quality measure for any subset S of $\mathcal{M}(G)$ such that $|S| = k$. The target is to find a set $S \subseteq \mathcal{M}(G)$ of size k that maximizes the quality measure. Examples include the following.

Problem 3.15 (TOP- k MAXIMAL CLIQUES BY SIZE). Given a graph G and a positive integer k , find a summary S of size k such that $\sum_{C \in S} |C|$ is maximized.

Problem 3.16 (SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE). Given a graph G and a positive integer k , find a summary S of size k such that the coverage size $|c(S)| = |\bigcup_{C \in S} C|$ is maximized.

For $k = 1$, [Problems 3.15](#) and [3.16](#) coincide with the problem of searching for the maximum clique of G .

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Fixed Size	Problem 3.15	TOP- k MAXIMAL CLIQUES BY SIZE <i>Summarization principles: size</i> Open
	Problem 3.16	SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE <i>Summarization principles: covering</i> [94, 127, 128, 129]
	Problem 3.17	SET OF k DISJOINT MAXIMAL CLIQUES MAXIMIZING COVERAGE <i>Summarization principles: internal overlap, covering</i> Open in general. $\sigma = 0$, non-maximal: [130, 131]
	Problem 3.18	SET OF k EDGE DISJOINT MAXIMAL CLIQUES MAXIMIZING COVERAGE <i>Summarization principles: internal overlap, covering</i> Open
	Problem 3.19	SET OF k MAXIMAL CLIQUES MINIMIZING ISOLATION <i>Summarization principles: isolation</i> Open

Table 3.3: An overview of Fixed Size problems discussed or proposed in [Section 3.4](#).

Problem 3.16, with the name TOP- k DIVERSIFIED MAXIMAL CLIQUES, has been studied in [94, 127, 128] which are surveyed in [Section 3.7.3](#). Additionally, it has been further addressed in [129, 132, 133, 134], considering a generalization in which vertices are weighted (see [Problem 3.21](#)).

Problem 3.17 (SET OF k DISJOINT MAXIMAL CLIQUES MAXIMIZING COVERAGE). Given a graph G , a positive integer k , and a value $\sigma \in [0, 1]$, find a σ -independent summary S of size k such that the coverage size $|c(S)| = |\bigcup_{C \in S} C|$ is maximized.

For $k = 1$, [Problem 3.17](#) coincides with the problem of searching for the maximum clique of G . When $\sigma = 0$, [Problem 3.17](#) has been studied in [130, 131] for non-necessarily maximal cliques with the name DISJOINT UNION OF CLIQUES.

Problem 3.18 (SET OF k EDGE DISJOINT MAXIMAL CLIQUES MAXIMIZING COVERAGE). Given a graph G , a positive integer k , and a value $\rho \in [0, 1]$, find a ρ -edge-independent summary S of size k such that the size of the coverage $|c(S)| = |\bigcup_{C \in S} C|$ of S is maximized.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

For $k = 1$, [Problem 3.18](#) coincides with the problem of searching for the maximum clique of G . Observe that any solution to [Problems 3.17](#) and [3.18](#) is also a solution to [Problem 3.16](#), while the converse does not necessarily hold.

Problem 3.19 (SET OF k MAXIMAL CLIQUES MINIMIZING ISOLATION). Given a graph G and a positive integer k , find a summary S of size k composed by ℓ -isolated maximal cliques such that the isolation factor ℓ with respect to an isolation concept (max-, avg-, min- ℓ -isolation) is minimized.

3.5 Variants of Summarization Principles and Problems

In addition to the classification of problems discussed in the previous section, several variants of summarization principles and problems can arise, depending for instance on the setting of the input graph or on the constraints that the desired summary must satisfy. This section explores some of these possible variants.

[Section 3.5.1](#) addresses the scenario where the input graph has weights assigned to vertices and edges. This yields a natural extension to problems grounded in the covering-based summarization principle, where the weight of the coverage is considered instead of its cardinality. [Section 3.5.2](#) focuses on the case where the input graph has probabilities assigned to both vertices and edges. This yields a natural extension to problems grounded in the size-based summarization principle, where the probability of existence of a maximal clique is also taken into account when computing the summary. [Section 3.5.3](#) examines probabilistic relaxations of the problems, i.e., variants where the summary must satisfy the desired property in an expected way.

An overview of the problems discussed in this section is provided in [Table 3.4](#), where they are grouped according to the four classes introduced in [Section 3.4](#). For each problem, we report the summarization principles, the specific variant considered, and the references where it has been studied.

3.5.1 Weighted Graphs

In a *vertex-weighted graph* $G = (V, E, w)$, each vertex of V is associated with a weight $w : V \rightarrow \mathbb{R}^+$. The meaningfulness of a clique in these application scenarios is given by the sum of the weights of its vertices rather than by their number. Therefore, a *maximal clique* is a clique whose weight is maximal. Fortunately, since the weight of each vertex is a positive number, a clique of maximal weight is also a maximal clique in terms of the number of its vertices

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Filtering	Problem 3.20	MAXIMAL CLIQUES BY WEIGHT <i>Summarization principles: (weighted) size</i> Variant: Weighted graphs Open
	Problem 3.23	α -MAXIMAL CLIQUES BY SIZE ON UNCERTAIN GRAPHS <i>Summarization principles: size</i> Variant: Uncertain graphs [135, 136, 137]
Fixed Size	Problem 3.21	SET OF k MAXIMAL CLIQUES MAXIMIZING WEIGHTED COVERAGE <i>Summarization principles: covering</i> Variant: Weighted graphs [129, 132, 133, 134]
Reduced Size	Problem 3.25	EXPECTED τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY <i>Summarization principles: external overlap</i> Variant: Probabilistic summary [87, 123]
Filtering + Fixed Size	Problem 3.22	TOP- k MAXIMAL CLIQUES ON UNCERTAIN GRAPHS <i>Summarization principles: size</i> Variant: Uncertain graphs [138]
	Problem 3.24	TOP- k α -MAXIMAL CLIQUES BY SIZE ON UNCERTAIN GRAPHS <i>Summarization principles: size</i> Variant: Uncertain graphs [139]

Table 3.4: An overview of the problems discussed or proposed in [Section 3.5](#).

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

and vice versa. However, the summarization principles listed in [Section 3.3](#) are affected by this change in perspective. For example, size-based summarization asks to focus on maximal cliques of large weight rather than large number of vertices. Consider the following reformulation of [Problem 3.1](#).

Problem 3.20 (MAXIMAL CLIQUES BY WEIGHT). Given a vertex-weighted graph G and a positive integer w , find all maximal cliques of weight at least w .

Given a summary $S \subseteq \mathcal{M}(G)$, the *weighted coverage* of S , denoted as $w(S)$, is defined as $w(S) = \sum_{v \in \bigcup_{C \in S} C} w(v)$ and is considered in the following problem.

Problem 3.21 (SET OF k MAXIMAL CLIQUES MAXIMIZING WEIGHTED COVERAGE). Given a vertex-weighted graph G and a positive integer k , find a summary S of size k , such that the weighted coverage $w(S)$ of S is maximized.

[Problem 3.21](#) is studied in [\[129, 132, 133, 134\]](#) under the name TOP k -DIVERSIFIED WEIGHTED MAXIMAL CLIQUES. Notice that the case of unweighted graphs can be viewed as a special case when each vertex has weight 1. For example, [Problem 3.21](#) generalizes [Problem 3.16](#), and an algorithm for the former can be applied also to the latter. Instead, when the parameter $k = 1$, [Problem 3.16](#) coincides with the maximum weight clique problem (see [\[140\]](#) and the references therein).

A similar extension can be considered by allowing edge-weighted graphs, although literature on this subject is available only for a few special cases [\[141, 142, 143\]](#).

3.5.2 Handling Uncertainty

An *uncertain graph* [\[138\]](#) is a graph $G = (V, E, \varphi, \psi)$, where $\varphi : V \rightarrow [0, 1]$ is a function assigning existence probability values to the vertices in V , and $\psi : E \rightarrow [0, 1]$ is a function assigning existence probability values to the edges in E upon the condition that both the endpoints of each edge exist. Therefore, a traditional graph, also referred to as an *exact graph*, can be considered as a special case of an uncertain graph, with all vertices and edges having an existence probability of 1. Usually, it is assumed that both the existence probabilities of vertices and the conditional existence probabilities of edges are mutually independent.

An exact graph $G' = (V', E')$ is *implied* by an uncertain graph $G = (V, E, \varphi, \psi)$, denoted by $G \Rightarrow G'$, if and only if $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

The *probability of G implicating G'* is

$$\begin{aligned} Pr(G \Rightarrow G') = & \prod_{v \in V'} \varphi(v) \cdot \prod_{v \in V \setminus V'} (1 - \varphi(v)) \cdot \prod_{(u,v) \in E'} \psi((u,v) \mid u, v) \cdot \\ & \cdot \prod_{(u,v) \in (E \cap (V' \times V')) \setminus E'} (1 - \psi((u,v) \mid u, v)) \end{aligned} \quad (3.3)$$

Let Ω denote the set of exact graphs implicated by an uncertain graph G . [Equation \(3.3\)](#) defines a probability distribution over all graphs in Ω . The probability $\pi_{max}(C)$ of a vertex set $C \subseteq V$ being a maximal clique across all implicated graphs G' in Ω is

$$\pi_{max}(C) = \sum_{G' \in \Omega \text{ and } G' \text{ contains } C \text{ as a maximal clique}} Pr(G \Rightarrow G') \quad (3.4)$$

This probability is referred to as the *probability of the maximal clique C* , i.e., the probability that C is a maximal clique in any implication of G . Moreover, given a collection S of vertex sets, the *sum of clique probabilities* of S , is defined as $\Sigma(S) = \sum_{C \in S} \pi_{max}(C)$. Building on the aforementioned definitions, the following problem has been discussed in the literature.

Problem 3.22 (TOP- k MAXIMAL CLIQUES ON UNCERTAIN GRAPHS). Given an uncertain graph G and two positive integers k and s , find a collection S of k vertex sets, such that each vertex set in S has size at least s and $\Sigma(S)$ is maximized.

When $k = |\mathcal{M}(G)|$, $s = 1$, and G is an exact graph, this problem coincides with maximal clique enumeration. [Problem 3.22](#) has been addressed in [\[138\]](#).

A restricted version of this model is when the uncertain graph has existence probabilities defined only for its edges (the existence probabilities of the vertices are always 1). This is discussed in [\[135, 136, 137, 139\]](#). In this case [Equation \(3.3\)](#) is simplified as follows

$$Pr(G \Rightarrow G') = \prod_{(u,v) \in E'} \psi((u,v)) \cdot \prod_{(u,v) \in E \setminus E'} (1 - \psi((u,v))) \quad (3.5)$$

Given an uncertain graph G where the existence probabilities of the vertices are equal to 1 and a set of vertices C , the *clique probability* of C , denoted by $\pi(C)$, is defined as the probability that in an exact graph implicated by G , C is a clique:

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

$$\pi(C) = \sum_{G' \in \Omega \text{ and } G' \text{ contains } C \text{ as a clique}} Pr(G \Rightarrow G') \quad (3.6)$$

which coincides with:

$$\pi(C) = \prod_{u,v \in C} \psi((u,v)) \quad (3.7)$$

Given a value $\alpha \in [0, 1]$, a vertex set C is an *α -clique* if $\pi(C) \geq \alpha$. Moreover, C is an *α -maximal clique* if it is an α -clique and there does not exist a vertex set $C' \supset C$ such that C' is an α -clique.

We mention the following problems from the literature.

Problem 3.23 (α -MAXIMAL CLIQUES BY SIZE ON UNCERTAIN GRAPHS). Given an uncertain graph G , a positive integer k and a value $\alpha \in [0, 1]$, find all α -maximal cliques of size at least k .

Problem 3.23 is addressed in [135, 136, 137] where it is referred to as (k, α) -MAXIMAL CLIQUES ON UNCERTAIN GRAPHS.

Problem 3.24 (TOP- k α -MAXIMAL CLIQUES BY SIZE ON UNCERTAIN GRAPHS). Given an uncertain graph G , a positive integer k and a value $\alpha \in [0, 1]$, find a set S of k α -maximal cliques such that $\sum_{C \in S} |C|$ is maximized.

Problem 3.24 is addressed in [139] where it is referred to as (k, α) -MAXIMAL CLIQUES ON UNCERTAIN GRAPHS.

Notice the differences between **Problem 3.22** and **Problem 3.24**. In the former, vertex sets with size at least s are ranked according to their probability, whereas in the latter, vertex sets corresponding to α -maximal cliques are ranked according to their size.

3.5.3 Probabilistic Relaxations

Given that computing an exact solution for most of the problems formulated above is challenging, we can relax the requirements by allowing the desired summary properties to be met by the algorithm's output with a certain probability. Till now, up to our knowledge, only the case of **Problem 3.9**, in which the external-overlap summarization principle is used, has been addressed in the literature.

Let $\mathcal{A}$ be a randomized algorithm that inserts a maximal clique $C \in \mathcal{M}(G)$ into its output summary S with probability $Pr[C \in S]$. The *expected visibility*

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

of a maximal clique $C \in \mathcal{M}(G)$ with respect to S , denoted as $\mathbb{E}[\mathcal{V}_S(C)]$, is defined as:

$$\mathbb{E}[\mathcal{V}_S(C)] = 1 \cdot \Pr[C \in S] + \mathcal{V}_S(C) \cdot \Pr[C \notin S] \quad (3.8)$$

Observe that, the output S of $\mathcal{A}$ is not known in advance and, hence, $\mathcal{V}_S(C)$ can be exactly determined only *a posteriori*. However, its value could be estimated and lower-bounded to guarantee that $\mathbb{E}[\mathcal{V}_S(C)]$ is also lower-bounded. A summary S of a graph G is *expected τ -visible*, with $0 \leq \tau \leq 1$, if the expected visibility of each maximal clique $C \in \mathcal{M}(G)$ with respect to S is at least τ , i.e. $\mathbb{E}[\mathcal{V}_S(C)] \geq \tau$.

Compare the following with [Problem 3.9](#).

Problem 3.25 (EXPECTED τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY). Given a graph G and a value τ , with $0 \leq \tau \leq 1$, find an expected τ -visible summary $S \subseteq \mathcal{M}(G)$ of small size.

As for [Problem 3.9](#), [Problem 3.25](#) coincides with the enumeration of maximal cliques when $\tau = 1$. [Problem 3.25](#) is addressed in [87, 123], which are surveyed in [Section 3.7.2](#).

3.5.4 Parameterized Approaches

Parameterized complexity [2, 3] offers a framework for addressing NP-hard problems by identifying a *parameter* k that captures some structural aspect of the problem. A problem is said to be *fixed-parameter tractable* (FPT) with respect to k if it can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$, where f is a computable function independent of the input size n . This allows for efficient algorithms when k is small, even for large instances. A common way to show that a problem is unlikely to be FPT with respect to a given parameter k is to give a parameterized reduction from a known W[1]-hard problem [2, 3], thereby establishing W[1]-hardness. A *parameterized reduction* reduces an instance (X, k) of a problem P that is W[1]-hard with respect to the parameter k to instance (X', k') of a problem P' in $f(k) \cdot |X|^{\mathcal{O}(1)}$ time such that k' only depends on k and such that (X, k) is a positive instance of P if and only if (X', k') is a positive instance of P' .

As mentioned in [Section 3.1](#), k -CLIQUE and, hence, [Problem 3.1](#), is known to be W[1]-hard with respect to the size k of the clique [2, 3]. To the best of our knowledge, among the summarization problems addressed in this chapter, an FPT approach has been proposed only for [Problem 3.2](#) [98, 99, 100] using

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

the isolation factor ℓ as parameter and leading to FPT algorithms to enumerate all ℓ -isolated maximal cliques (see [Section 3.7.4](#)).

While any structural graph parameter could, in principle, be a candidate for FPT analysis, it is particularly interesting to consider parameters that are explicitly present in the problem definition itself. Unfortunately, most parameters that appear in the problems listed in [Section 3.4](#), such as τ , σ , ρ , and c , are real numbers in $[0, 1]$ and, hence, cannot be directly used as parameters for an FPT reduction or algorithmic approach. To deal with this issue, a strategy could be that of addressing the equivalent problems that consider absolute values instead of percentages. For instance, in [Problem 3.9](#) instead of considering the local visibility of C with respect to C' as $\mathcal{V}_{C'}(C) = \frac{|C \cap C'|}{|C|}$ one could consider the measure $|C \cap C'|$. This would yield a new visibility definition and a new formulation for [Problem 3.9](#) using an integer parameter that could be exploited for an FPT analysis.

3.6 Research Targets for Summarization Problems

In [Sections 3.4](#) and [3.5](#) we introduced and classified several problems falling in the broad research area of maximal clique summarization. The main research target is that of devising algorithms to find solutions to such problems. However, specific problems could be addressed also from different perspectives, either to better understand the underlying complexity of the questions or to propose solutions tailored to particular application contexts. Such alternative research targets include devising exact or enumerative algorithms, investigating hardness or W[1]-hardness, conceiving FPT approaches, applying probabilistic techniques, generalizing to weighted graphs or focusing on particular graph families, extending the results to graph with uncertainty, proposing approximated solutions. In this section we highlight some promising research directions based on the classification introduced in [Section 3.4](#).

Filtering problems. Since Filtering problems ask to evaluate a Boolean property independently for each maximal clique, these problems are well-suited for enumeration algorithms, while probabilistic approaches do not appear to be particularly meaningful.

For [Problem 3.1](#) (MAXIMAL CLIQUES BY SIZE), related to the size of maximal cliques, known results on W[1]-hardness [\[2, 3\]](#) imply that designing FPT algorithms with respect to the size of the cliques is likely to be difficult. Instead, the research could focus on developing efficient enumerative algorithms, either with output-sensitive running time or with bounded-delay guarantees. When

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

$s = \omega(G)$, randomized approaches for [Problem 3.1](#) (MAXIMAL CLIQUES BY SIZE) have been presented in [144]. An alternative definition of clique size could be adopted for weighted input graphs, leading to a weighted variant of the problem (see [Problem 3.20](#)).

Regarding [Problem 3.2](#) (ℓ -ISOLATED MAXIMAL CLIQUES), which focuses on the isolation of maximal cliques within the summary, positive results already exist in terms of FPT algorithms [98, 99, 100].

Search and Reduced Size problems. Search and Reduced Size problems are both defined through a Boolean property that has to be satisfied by the summary as a whole. Therefore, they often share common research goals. For several of these problems, the literature provides NP-hardness results, at least for specific values of the parameter. For example: [Problem 3.9](#) (τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) coincides with the enumeration of the maximal cliques when $\tau = 1$, which implies the complexity bounds of the maximum clique problem discussed in [Section 3.1](#). When $c = 1$, [Problem 3.10](#) (c -COVERING MAXIMAL CLIQUE SMALL SUMMARY) is usually called CLIQUE COVER. Such a problem can have four different formulations, depending on whether the cliques are requested to be maximal and whether they can possibly intersect. The NP-hardness proof originally given by Karp [1, Problem 13] applies to non-maximal cliques, either disjoint (also called PARTITION INTO CLIQUES [145, Problem GT15]) or possibly intersecting. The problem of finding a small set of intersecting covering cliques can be easily reduced to that of finding covering intersecting maximal cliques and vice versa. Therefore, the latter problem is also NP-hard. To the best of our knowledge the complexity of finding a small set of covering maximal disjoint cliques is unknown. Observe that this problem coincides with [Problem 3.14](#) (c -COVERING σ -INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY) when $c = 1$ and $\sigma = 0$. The existence version corresponds, instead, to [Problem 3.5](#) (FIND ANY c -COVERING σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY). [Problem 3.6](#) (FIND ANY c -EDGE-COVERING σ -INDEPENDENT MAXIMAL CLIQUE SUMMARY) when $c = 1$ and $\sigma = 0$ is proven to be NP-hard in [107].

For Reduced Size problems, where the target is finding a “small” summary satisfying a given property, it makes sense considering approximations of the minimum size summary. In this respect, very few results are known. For example, it can be proved that [Problem 3.10](#) (c -COVERING MAXIMAL CLIQUE SMALL SUMMARY) for $c = 1$ does not admit, unless $P = NP$, a polynomial time approximation algorithm for any $\epsilon > 0$ that, on n -vertex graphs, achieves an approximation ratio better than $n^{1-\epsilon}$ on the number of maximal cliques covering the vertices of G . Indeed, the polynomial reduction from GRAPH COLORING to

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

CLIQUE COVER provided by [1] maps the number of colors of the coloring to the number of (maximal) cliques covering the vertices of G . Therefore, the same hardness of approximation results for the former problem proved in [5] applies to the latter.

Algorithmic FPT approaches could be devised using as FPT parameters the specific values in the problems definition (e.g., τ , σ , ρ , and c), as described in Section 3.5.4.

Probabilistic approaches could also prove useful in this setting, since the Boolean property is evaluated over the entire summary and the output property might be satisfied with a certain probability. Only Problem 3.25 (EXPECTED τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) has been addressed using a probabilistic algorithm [87, 123].

Fixed Size problems. All Fixed Size problems depend on a parameter k and, when $k = 1$, coincide with the search for one of the maximal cliques that maximize the objective function. Very often, these search problems are known to be NP-hard. For example, as noted in [94], Problem 3.15 (TOP- k MAXIMAL CLIQUES BY SIZE) and Problem 3.16 (SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE) when $k = 1$ coincide with MAXIMUM CLIQUE. The NP-hardness of Problem 3.16 for an arbitrary k has been claimed in [127, 128]. We remark that MAXIMUM CLIQUE admits an FPT algorithm in the degeneracy of the graph [146].

For Fixed Size problems, approximation approaches seem extremely challenging: in many cases it is not difficult to construct instances such that any k -summary that does not fully satisfy the problem condition is arbitrarily far from the optimum. For the special case of MAXIMUM CLIQUE, which, as said, is hard to approximate within a factor of $n^{1-\epsilon}$ [4, 5], the best known approximation factor is $\mathcal{O}(n(\log \log n)^2/(\log n)^3)$ [147]. We refer to [148, 149] for surveys on FPT approaches, W[1]-hardness, and approximation of MAXIMUM CLIQUE and its relaxations.

3.7 A Review of Summarization Algorithms

In addition to outlining key problems and summarization principles, Tables 3.1 to 3.3 provide references to previous studies that addressed these problems, also highlighting which of them are still open. In this section we provide an in-depth review of algorithmic advancements, using the summarization principles introduced in Section 3.3 as a first level of classification.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

3.7.1 Size-based Summarization

Detecting a maximum clique. When $k = 1$, both [Problem 3.15](#) (TOP- k MAXIMAL CLIQUES BY SIZE) and [Problem 3.16](#) (SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE) coincide with the problem of detecting a maximum clique. This is one of the most studied NP-hard problems. Several surveys on the maximum clique problem are available in the literature, including [\[150, 151, 152\]](#). In [\[152\]](#), which is, to the best of our knowledge, the most recent, exact and heuristic approaches are considered in detail, and inapproximability results and parametric complexity completeness are briefly discussed. Benchmarks and experimental comparisons as well as generalizations to vertex-weighted and edge-weighted graphs are also discussed.

Enumerating maximum cliques. [Problem 3.1](#) (MAXIMAL CLIQUES BY SIZE) is equivalent to the problem of enumerating all maximum cliques when only cliques of size $\omega(G)$ should be included in the summary, where $\omega(G)$ is the *clique number* of G , i.e., the number of vertices in any maximum clique. This problem has been addressed in [\[153\]](#) and in [\[144\]](#), providing exact and probabilistic solutions, respectively.

Enumerating large maximal cliques. In its general formulation, [Problem 3.1](#) is addressed in [\[101, 102\]](#), which propose an algorithmic approach that, before launching the enumeration of the large maximal cliques, performs two filtering steps to the input graph.

Since the summary S must contain only maximal cliques of size at least s , a vertex $v \in V$ cannot belong in any maximal clique of S unless its degree is at least $s - 1$. The first filtering step recursively removes vertices based on this observation, actually computing the s -core of the input graph [\[154\]](#), an operation that can be efficiently performed [\[155\]](#).

An edge $e = (u, v)$ can participate to a maximal clique of size at least s only if both u and v have at least $s - 1$ neighbors in common, i.e., $|N(u) \cap N(v)| \geq s - 1$. Moreover, any vertex v in a maximal clique of size at least s must have at least $s - 1$ neighbors, such that for each vertex u of them, it holds that $|N(v) \cap N(u)| \geq s - 1$. Considering these two observations, the graph is filtered again by first removing edges and then removing vertices that cannot belong to a maximal clique of size at least s .

The enumeration algorithm explores the recursion tree of [Algorithm 1.1](#) by means of a breadth-first search instead of a depth-first-search. Furthermore, when a vertex v is not added to the current solution, a *tabu list* is created with all the vertices in T that are not adjacent to v . The tabu lists are used to stop

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

the exploration of a branch of the recursion tree when all the maximal cliques generated by that branch are guaranteed to have size less than s .

3.7.2 Overlap-based Summarization

Problem 3.9 (τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) and its probabilistic relaxation **Problem 3.25** (EXPECTED τ -VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) are addressed in [87, 123, 124, 156], where τ -visible summaries are computed by a modified **Algorithm 1.1** approach discussed in **Chapter 1**. In particular, while traversing the **Algorithm 1.1** recursive call tree, these approaches perform two kinds of operations:

RETENTION: when a maximal clique C is found a decision is taken about whether adding C to S or not;

PRUNING: when the current recursive call corresponds to a non-maximal clique a decision is taken about whether pruning the current recursion subtree, so reducing the considered solution space.

It can be observed that **Algorithm 1.1** discussed in **Chapter 1** yields an enumeration in which similar cliques tend to be near in the output sequence. Based on this observation, both the above introduced operations can be accomplished by considering the last maximal clique C' added to S as a representative for all the cliques in S . In this case, **RETENTION** does not necessarily filter out all maximal cliques that have a visibility greater than τ with respect to maximal cliques already added to S , i.e., the produced summary is not guaranteed to be minimal. When performing **PRUNING**, instead, considering C' only does not affect the produced summary but only the computation time.

Pruning by lower bounding visibility. The **PRUNING** operation, as defined in [87], determines whether the subtree $\mathcal{T}_c$ of the complete search tree $\mathcal{T}_G$ from **Algorithm 1.1**, generated when the algorithm is executed on a graph G , and rooted at the current node c corresponding to the clique C , may contain a maximal clique with low visibility with respect to C' . In particular, let $\underline{r}$ be a lower-bound on the visibility of the maximal cliques contained in $\mathcal{T}_c$ with respect to C' , the current branch of the computation is pruned if $\underline{r} \geq \tau$. In order to compute $\underline{r}$, one needs to estimate the maximum size of a maximal clique D corresponding to a leaf x of $\mathcal{T}_c$, which is the size of C plus the depth d of $\mathcal{T}_c$, i.e., $|D| = |C| + d$. Second, one needs to estimate the size of the intersection between D and C' , which may be given by $|C \cap C'|$ plus the number of vertices in $D \setminus C$ that are not in C' . Let d_y be the number of vertices of D that are

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

not in C' . Then the size of the intersection $|D \cap C'| = |C \cap C'| + (d - d_y)$. Therefore, the lower-bound $\underline{r}$ on the visibility of D with respect to C' can be formulated as follows.

$$\underline{r} = \min_{1 \leq d \leq \bar{d}} \frac{|C \cap C'| + \max\{d - \bar{d}_y, 0\}}{|C| + d} \quad (3.9)$$

where $\bar{d}$ is an upper-bound on the depth d of $\mathcal{T}_C$; $\bar{d}_y$ is an upper-bound on d_y ; and the max function is needed since $d - \bar{d}_y$ could be negative. It remains to discuss how to estimate $\bar{d}$ and $\bar{d}_y$. Recall that the set of candidate vertices that can possibly be added to C is denoted by P . Observe that an upper bound for d coincides with an upper bound for the maximum clique in $G[P]$. Hence, a trivial upper bound could be to consider the size of the candidate set P . In [87] three heuristics are proposed to upper bound d : (i) the maximum degree $\bar{d}_\Delta$ of any vertex in $G[P]$, (ii) the maximum value $\bar{d}_h$ of h such that there are h vertices with at least degree $h - 1$ in $G[P]$, and (iii) the maximum core number $\bar{d}_{core}$ of $G[P]$. A fourth upper bound for d is provided in [156]: (iv) $\bar{d}_{truss}$ that is the maximum truss number of $G[P]$. Observe that:

$$\bar{d}_\Delta \geq \bar{d}_h \geq \bar{d}_{core} \geq \bar{d}_{truss} \quad (3.10)$$

The first inequality is trivial, the second one holds because $\bar{d}_{core}$ requires the h vertices to be connected and the third holds since a k -truss must be a $(k - 1)$ -core. Computational complexities for the four approaches above are, respectively:

$$\mathcal{O}(1) \leq \mathcal{O}(|V[P]|) \leq \mathcal{O}(|E[P]|) \leq \mathcal{O}(|E[P]|^{1.5}) \quad (3.11)$$

$\bar{d}_y$ can be estimated with $|P \setminus C'|$, or with the number of vertices in $P \setminus C'$ having degree at least d , or simply as the value of d .

Avoiding costly pruning operations. Even if the approach described above improves the running time with respect to the classic maximal clique enumeration algorithm, it still suffers from two main drawbacks: (a) it requires a new costly estimation of $\bar{d}$ independently at each recursion and (b) it tries to prune several times subtrees that are not prunable because of some common leaves. Observe that drawback (b) is also costly, since each pruning attempt needs an estimation of $\bar{r}$ which in turn requires an estimation of $\bar{d}$. To address these limitations, in [124] three novel keeping strategies are proposed to determine when a subtree is not prunable with the goal to avoid unnecessary bound estimation. In practice, these strategies consists of upper bounding strategies of visibility in contrast with the original lower bound approach exploited during

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

the pruning attempt. Moreover, instead to calculate $\bar{d}$ from zero, an update approach is proposed to adjust the $\bar{d}$ from the parent recursion denoted by $\bar{d}_0$. More formally, each time before entering a new recursion with the current partial clique C , if the following inequality holds:

$$\frac{|C \cap C'|}{|C| + \bar{d}_0 - 1} \geq \tau \quad (3.12)$$

the current subtree can be pruned. Notice that this pruning strategy requires only $\mathcal{O}(1)$ extra time complexity.

We now discuss the keeping strategies. All of them are evaluated at each recursive step. If one of the following inequalities holds then the current subtree will be entirely explored without checking the pruning conditions. The inequalities guarantee that any maximal clique D generated from C cannot be witnessed by the last reported maximal clique C' in S . The first strategy is by *size filtering*

$$\frac{|C'|}{|C| + 1} < \tau \quad (3.13)$$

with a $\mathcal{O}(1)$ time complexity. The second one is by *bound reusing*

$$\frac{|C \cap C'| + \bar{d}_0 - 1}{|C| + \bar{d}_0 - 1} < \tau \quad (3.14)$$

with a $\mathcal{O}(1)$ time complexity. The last one is by *candidate tracing*

$$\frac{|C \cap C'| + |T \cap C'|}{|C| + 1} < \tau \quad (3.15)$$

with a time complexity of $\mathcal{O}(\kappa(G)/|D|)$ if degeneracy ordering is performed [8], where $\kappa(G)$ is the degeneracy of G .

Putting it all together, the original approach can be refined processing sequentially the pruning and keeping strategies discussed before. In this way, only when all of them fail, the costly inequality $\bar{r} \geq \tau$ is evaluated.

A global retention approach. Instead of considering the visibility with respect to the last summarized clique, a different approach to remove redundancy might exploit a global view of summary S . In more details, when evaluating a maximal clique C , we might compare C against all the cliques in S . This is however a costly operation that requires an efficient implementation. A possible strategy is presented in [87]: clique C and any other clique of S are further compared only if the intersection of their vertex id ranges is nonempty.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Probabilistic method. In the randomized approach introduced in [87], RETENTION is performed with probability $1 - \sqrt[l]{s(\underline{r})}$, where $\bar{l}$ is the denominator of Equation (3.9), i.e., $\bar{l} = \bar{d} + |C|$, while PRUNING is performed with probability $\sqrt[l^C]{s(\underline{r})}$, where $s(\underline{r})$ is a sampling probability function

$$s(\underline{r}) = \frac{(1 - r)(2 - \tau)}{(2 - r - \tau)} \quad (3.16)$$

that corresponds to a monotonically decreasing function with the property to have value 1 when $r = 0$ and value 0 when $r = 1$. Therefore, the larger is the estimated value of $\underline{r}$, the higher is the chance to effectively prune $\mathcal{T}_c$. Moreover, the larger is $\bar{d}$ (and thus $\bar{l}$) the larger is the number of recursion calls and the smaller is the probability of pruning $\mathcal{T}_c$.

In [123], a new sampling probability function is proposed to overcome two main issues on Equation (3.16). The first is that when $r \in [\tau, 1]$, we always obtain $s(\underline{r}) > 0$. This means that there is a positive probability to add the maximal clique into the summary, even if it is τ -visible with respect to S . The second issue is that when $r = 0$, we have $s(\underline{r}) = 1$. In this situation, even if the newly generated maximal clique has vertices that are not covered in S , it could be covered by some future maximal cliques. So, it makes sense to have a non-null probability to sample the clique, especially considering that the aim is to compute an expected τ -visible summary. The sampling probability function proposed by [123] is

$$s_{opt}(\underline{r}) = \begin{cases} \frac{\tau - r}{1 - r}, & \text{if } r \in [0, \tau) \\ 0, & \text{if } r \in [\tau, 1] \end{cases} \quad (3.17)$$

It can be proved [123] that the adoption of function $s_{opt}(\underline{r})$ guarantees to compute an expected τ -visible summary and that $s_{opt}(\underline{r}) \leq s(\underline{r})$, $\forall r \in [0, 1]$, resulting in a more effective pruning strategy.

3.7.3 Covering-based Summarization

Problem 3.16 (SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE) has been addressed in [94, 127, 128] using very different approaches: a variation of Algorithm 1.1 for the maximal clique enumeration [94]; a configuration checking strategy [127]; and a strategy based on formal concept analysis [128].

Approaches based on maximal clique enumeration. The approach proposed in [94] to find a summary S is based on a simple change to the maximal clique enumeration algorithm described in Chapter 1 (Algorithm 1.1),

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

where at Line 4 a maximal clique C is not necessarily reported. In order to describe this approach we introduce some definitions. Given a summary $S \subseteq \mathcal{M}(G)$ of maximal cliques of G , the *private-vertex-set* for each clique $C \in S$, denoted by $p(C, S)$, is the set of vertices in C that are not contained in any other maximal clique in S , i.e., $p(C, S) = C \setminus c(S \setminus \{C\})$. Intuitively, $p(C, S)$ is the set of vertices of G that would be removed from $c(S)$ if C was removed from S . Each vertex $v \in p(C, S)$ is called a *private vertex of C in S* . The *min-cover-clique of S* , denoted by $C_m(S)$, is the clique $C \in S$ with minimum private-vertex-set size $|p(C, S)|$, i.e., $C_m(S) = \arg \min_{C \in S} \{|p(C, S)|\}$. If more than one maximal clique of S has minimum $|p(C, S)|$ then one such clique is arbitrarily chosen to be the min-cover-clique of S . Intuitively, the min-cover-clique of S is the clique that reduces the coverage size $|c(S)|$ by the minimum possible when removed from S .

During the enumeration process, the first k maximal cliques are added to summary S and, upon discovering a new maximal clique C , the min-cover-clique $C_m(S)$ is replaced by C when this yields a summary S' with a sufficiently larger coverage, that is, when the private-vertex-set size of C in S' is such that $|p(C, S')| > |p(C_m(S), S)| + \alpha \cdot \frac{|c(S)|}{|S|}$, where α is a parameter ($0 \leq \alpha \leq 1$). In [94] it is shown that, when $\alpha = 1$, the above approach computes a summary S , such that $|c(S)| \geq 0.25 \cdot |c(S^*)|$, where S^* is the optimal summary, i.e., the set of k maximal cliques that maximizes coverage.

Pursuing efficiency. A naïve implementation of the above described approach would be extremely inefficient, since each update of S to S' implies computing the min-cover-clique $C_m(S)$ of S , the size of the private-vertex-set of C in S' , and the coverage size $|c(S)|$. Overall, this requires computing again the private-vertex-set of each maximal clique in S in the worst case. A data structure to efficiently address this problem is presented in [94]. This is based on the concept of ‘reverse coverage’ for vertices, i.e., the cliques of the summary S that a vertex belongs to. The reverse coverage is exploited to update a data structure that makes it possible to retrieve the aforementioned quantities (min-cover-clique, private-vertex-set size and coverage size) without the need of recomputing them from scratch.

It is proved in [94] that this implementation, while using only $\mathcal{O}(\sum_{C \in S} |C|)$ space, can produce a summary in the same time required for enumerating all the maximal cliques in G , in short, without requiring any extra asymptotic costs with respect to both time and space complexities.

On top of this approach, as shown in [94], local and global pruning strategies of the recursion tree can be efficiently implemented to reduce the search space.

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

Global pruning prioritizes the expansion of the current clique towards vertices with a higher ‘appetibility’ score. This has the effect of allowing to prune those subtrees that correspond to low ‘appetibility’ score and that have no potential to be expanded to improve the quality of the current candidate. Local pruning is based on an upper-bound estimation of the potential of the maximal cliques contained into the subtree rooted at the current clique. Adding pruning does not increase the asymptotic running time but yields a substantial boost on the computation times in practice.

A further strategy used in [94] to improve efficiency is to start from a suitable set of initial candidate maximal cliques of S such that both global and local pruning conditions can be satisfied earlier.

Approaches based on configuration checking strategy. Another approach to solve [Problem 3.16](#) is proposed in [127] and it is based on a variant of *Configuration Checking* (CC) [157] combined with an application of the *Best from Multiple Selection* (BMS) [158]. Configuration Checking is a simple strategy used to reduce the cycling phenomenon in a local search when solving combinatorial optimization problems. In this case, it is used to sample the solution space maintaining a tabu set to enforce the exploration of different portions of it. Since CC is not applicable to [Problem 3.16](#), a variant of it named *Enhanced Configuration Checking* (ECC) is proposed in [127]. The idea of ECC is that a maximal clique can be added to S only if it does not contain tabu vertices.

1. When a maximal clique C is removed from S , every vertex $v \in p(C, S)$ is added to the tabu set and every tabu vertex u neighbor of v is removed from the tabu set.
2. When a maximal clique C is added to S , every tabu vertex neighbor of a vertex $v \in p(C, S)$ is removed from the tabu set.

A Boolean array *ConfChange* can be used to implement the configuration checking strategy: each element of array *ConfChange* refers to a vertex v and has value 0 if the vertex belongs to the tabu set and 1, otherwise. At the beginning each vertex $v \in V$ has *ConfChange*[v] = 1. Given a set of maximal cliques S , the *configuration* of a vertex $v \in V$ is the set $conf(v) = \{u | u \in N(v) \setminus c(S)\}$. Observe that *ConfChange*[v] = 1 either if v has never been considered so far or if a neighbor of v has been removed or added to $c(S)$, that is, if its configuration changed.

A clique C is first constructed starting from a random vertex v having *ConfChange*[v] = 1. A *candidate set* T is maintained, consisting of non-tabu

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

neighbors of all vertices in C . Each vertex in T is associated with a benefit measure $\hat{b}[v] = |N(v) \cap T|$. While T is non-empty one vertex $v \in T$ is moved to C using the BMS strategy. If T is sufficiently small ($|T| < m$, where m is a suitable parameter), v is greedily chosen based on its benefit. Otherwise, v is selected among m random vertices of T maximizing the benefit. Observe that, the greater is m the higher is the probability of having a summary with high coverage, at the cost of a higher computation time. Using the above procedure the generated clique C may not be maximal in G as it could be contained into a larger clique involving tabu vertices. Therefore, in a post-processing phase, clique C is extended to a maximal one by iteratively adding tabu vertices adjacent to all vertices in C [159].

The general approach constructs several candidate summaries until a cut-off time is reached, choosing the one with highest coverage. The candidate summaries are computed using different values of m . Each candidate summary is built by adding k maximal cliques computed with the above described strategy, and then refining the solution for a bounded number of times by trying to replace the maximal clique C in S with smaller $|p(C, S)|$ with some other clique that improves $|c(S)|$.

Approaches based on formal concept analysis. A completely different approach to solve Problem 3.16 is proposed in [128] casting the problem into the framework of formal concept analysis (FCA), where the maximal cliques of a graph G correspond to equiconcepts in the formal concept lattice of G . The algorithm is based on a strategy that leverages FCA to find k -cliques originally proposed in [160].

In FCA, a triple $K = (O, A, I)$ is called a *formal context* if O and A are sets and $I \subseteq O \times A$. Elements of O are called *objects*, those of A are called *attributes*, whereas I is the *incidence* of the context K . For $X \subseteq O$, we define $X' = \{a \in A \mid \forall o \in X : (o, a) \in I\}$. Analogously, for $Y \subseteq A$, we define $Y' = \{o \in O \mid \forall a \in Y : (o, a) \in I\}$. A pair $H = (X, Y)$ is a *formal concept* of K if and only if $X \subseteq O$, $Y \subseteq A$, $X' = Y$, and $X = Y'$. Formal concepts of a given formal context are ordered by a partial ordering ' $\leq$ ' defined as follows: $(X_1, Y_1) \leq (X_2, Y_2) \iff X_1 \subseteq X_2 \text{ or } Y_2 \subseteq Y_1$. Let $C(K)$ denote the set of all formal concepts of K , then ' $\leq$ ' is a partial relation of $C(K)$. A concept lattice $L = (C(K), \leq)$ can be obtained by all elements in $C(K)$ of a context K with the partial order $\leq$. An *equiconcept* is a special concept $H = (X, Y) \in C(K)$, such that $X = Y$. We denote by $EC(K)$ the set of all equiconcepts of $C(K)$. Finally, given three equiconcepts $H_1 = (X_1, Y_1)$, $H_2 = (X_2, Y_2)$ and $H_3 = (X_3, Y_3) \in EC(K)$, such that $H_1 \leq H_3$ and $H_2 \leq H_3$,

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

then H_3 is denoted as the *father* concept of H_1 and H_2 , and H_1 is denoted as the *brother* concept of H_2 .

The *formal context K of a graph $G = (V, E)$* is constructed by choosing $O = A = V$ and, for every $(v_i, v_j) \in E$, both $(v_i, v_j) \in I$ and $(v_j, v_i) \in I$. Also, for every $v_i \in V$, $(v_i, v_i) \in I$.

It is proved in [128, Theorem 1] that $\mathcal{M}(G) = EC(K)$, i.e., the set $\mathcal{M}(G)$ of maximal cliques of G corresponds to the set $EC(K)$ of equiconcepts in the formal concept lattice $L = (C(K), \leq)$ obtained from the formal context K of G .

In order to obtain a summary S , $EC(K)$ is computed in a similar way as described in [160]. The procedure described in [128] adds k elements to S following an iterative approach where, at each step, the clique C corresponding to the equiconcept $H = (X, X)$ with the largest $|X|$ is added to S . After the addition, in order to pursue diversification, both H and all its brother equiconcepts are removed from $EC(K)$.

3.7.4 Isolation-based Summarization

In this section, we review the results from the literature concerning **Problem 3.2** (ℓ -ISOLATED MAXIMAL CLIQUES) that has been studied in [98, 99, 100]. We first discuss the approach proposed in [98] to obtain a summary S that contains all avg- ℓ -isolated maximal cliques in a given graph $G = (V, E)$ with n vertices and m edges in $\mathcal{O}(\ell^4 \cdot 2^{2\ell} \cdot m)$ time. Then, we discuss the approach proposed in [99], which shows how to solve **Problem 3.2** for the two other isolation concepts: when S must contain all max- ℓ -isolated maximal cliques or all min- ℓ -isolated maximal cliques.

In the following, we assume that the vertices of G are sorted by their degree such that $u < v \Rightarrow \delta(u) \leq \delta(v)$, meaning that we associate an *index* to each vertex. Let $N[v] = N(v) \cup \{v\}$. Let $N_+[v] = \{u \in N[v] \mid u > v\}$ and $N_-(v) = \{u \in N(v) \mid u < v\}$. Let $k = |N[v]|$. For any avg- ℓ -isolated maximal clique C , there is a vertex in C , referred to as a *pivot*, that has less than ℓ edges outgoing from C . Observe that, an avg- ℓ -isolated maximal clique may have more than one pivot. Using the vertex order, we can select the pivot to be the vertex in the maximal clique having the minimum index.

The general idea discussed in [98] to find the summary S is to scan each vertex v and enumerate all avg- ℓ -isolated maximal cliques whose pivots are v , by removing at most $\ell - 1$ vertices from $N(v)$. For every vertex v , three main steps are performed, which are discussed in the following.

Trimming stage. First, if $|N_-(v)| > \ell - 1$, vertex v can safely be discarded as it cannot be a pivot of any avg- ℓ -isolated maximal clique. Otherwise, a candidate

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

set $C \subseteq N[v]$ is defined as $C = N[v] \setminus N_-(v) = \{v = u_1, \dots, u_k\}$. The idea is to remove from C all vertices that cannot belong to an avg- ℓ -isolated maximal clique with v as the pivot. At the end, C contains all avg- ℓ -isolated maximal cliques with v as the pivot. Additionally, let $C^{u_i} = \{w \in C \mid w < u_i\} \cup \{u_i\}$. The following conditions are then considered:

- (a) $\delta(u_i) < (\ell + 1) \cdot |C| - 1$;
- (b) u_i has less than $\ell \cdot |C|$ outgoing edges from C ;
- (c) u_i has at least $|C| - \ell$ neighbors in C ;
- (d) C^{u_i} has less than $\ell \cdot (\ell + 1) \cdot |C^{u_i}|$ outgoing edges.

For each vertex u_i , where $i = 1, \dots, k$, condition (a) is tested and u_i is eventually removed from C . Next, conditions (b) – (d) are checked for each vertex u_i , where $i = 2, \dots, k$, and u_i is eventually removed from C . At any time, if the number of deleted vertices exceeds $\ell - 1$ (equivalently, if $|C| < k - \ell - 1$), then v cannot be a pivot and the entire procedure is terminated early.

Condition (a) ensures that high-degree vertices are early removed. Then, the other three conditions are evaluated only on small-degree vertices.

Enumerating stage. After the trimming stage, let k' denote the size of the candidate set C , with $k - \ell < k' \leq k$. At this point, C is a superset of all avg- ℓ -isolated maximal cliques with pivot v . The goal is to find maximal cliques by removing at most $\ell' = \ell - 1 - (k - k')$ vertices from C . Observe that, if $C' \subseteq C$ is a clique, then $C \setminus C'$ forms a vertex cover in the complement graph $\overline{G[C]}$. Using this property, listing all maximal cliques in $G[C]$ that can be obtained by removing at most ℓ' vertices is equivalent to enumerating minimal vertex covers with at most ℓ' vertices in $\overline{G[C]}$. There exists an algorithm for this problem with a time complexity of $\mathcal{O}(2^{\ell'} \cdot \overline{m})$ [3], where $\overline{m}$ is the number of edges of $\overline{G[C]}$. We denote as Q_C the set of maximal cliques obtained at the end of this stage.

Screening stage. This final step aims to test whether elements in Q_C are indeed avg- ℓ -isolated maximal cliques. Regarding isolation, each $Q \in Q_C$ can be efficiently tested to determine if it is avg- ℓ -isolated; eventually Q is removed from Q_C . Regarding maximality, a simpler check is introduced in [99]. Since high-degree vertices are removed before the enumeration stage, all vertices in $N[v] \setminus Q$ must be considered, as one of these vertices could potentially be a common neighbor of all vertices in Q . Therefore, for each deleted vertex w from

3. MAXIMAL CLIQUES SUMMARIZATION: PRINCIPLES, PROBLEM CLASSIFICATION, AND ALGORITHMIC APPROACHES

$N[v] \setminus C$, it is checked whether w is a neighbor of every vertex in Q . If so, Q is discarded. Otherwise, Q is added to S .

Other isolation concepts. We now discuss how the procedure described above can be adapted [99] to obtain a summary S that must contain either all min- ℓ -isolated maximal cliques or all max- ℓ -isolated maximal cliques. For both the isolation concepts, the enumeration stage remains unchanged.

When considering min- ℓ -isolated maximal cliques, recall that only maximal cliques with at least one vertex having less than ℓ neighbors outside the clique are relevant. During the trimming stage, vertices with few neighbors in the candidate set C must be removed because including them would result in a maximal clique where the pivot vertex v has at least ℓ outgoing edges. Only a single condition has to be checked for each vertex u_i :

(a') u_i has at least $|C| - \ell$ neighbors in C .

In the screening stage, only the vertices in $N_-(v)$ must be considered, as the enumerated cliques are maximal in $G[N_+[v]]$. For each $w \in N_-(v)$ and for each $Q \in Q_C$, it is tested whether $Q \subseteq N(w)$. If so, Q is discarded. Otherwise, Q is added to S .

When considering max- ℓ -isolated maximal cliques, recall that only maximal cliques in which every vertex has less than ℓ neighbors outside the clique are relevant. In the trimming stage, high-degree vertices are removed as in the case of ℓ -isolated maximal cliques, but since the max-isolation concept is stricter, the degree of the vertices is even more restricted compared to avg-isolation. Two conditions are checked for each vertex u_i :

(a'') $\delta(u_i) < |C| + \ell - 1$;

(b'') u_i has at least $|C| - \ell$ neighbors in C .

The screening stage remains unchanged with respect to avg-isolation.

Filtering the Clique Explosion: Enumerating ℓ -Isolated Maximal Cliques Efficiently

4.1 Introduction

Cliques capture dense, highly-connected network structures. Their enumeration is important in many application domains (see [Section 3.1](#)). In particular, enumerating maximal cliques helps reveal the graph's structure by uncovering tightly connected groups of vertices that share meaningful relationships.

In general, the number of maximal cliques in an n -vertex graph can be exponential in n [[13](#)]. Hence, not only their enumeration requires considerable time, but navigating such a large number of structures is also difficult. Imposing additional constraints may pinpoint a subset of domain-relevant structures, substantially reduce the number of enumerated maximal cliques, and enable more effective use (see [Chapter 3](#)).

In settings where loosely connected cliques are more informative than ones tightly embedded in the surrounding network, the notion of ℓ -isolation can be especially useful. Under ℓ -isolation, vertices of the clique are required to have fewer than ℓ edges to vertices outside the clique (see [Section 3.3.4](#)). In more details, one may require that the number of edges leaving a maximal clique of size k does not exceed $\ell \cdot k$ (called *avg- ℓ -isolation*) [[161](#), [162](#)] or that the maximum external degree of each vertex of the maximal clique is at most ℓ

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

(called *max- ℓ -isolation*) [100].

In [161, 162] and [100], FPT algorithms in the isolation factor ℓ have been proposed to enumerate avg- ℓ -isolated and max- ℓ -isolated maximal cliques, respectively. It was also shown in [161, 162] that ℓ must be constant to achieve linear-time enumeration, since otherwise one can construct instances admitting a superlinear number of avg- ℓ -isolated maximal cliques.

A systematic comparison of different isolation notions, complemented by an experimental evaluation on synthetic and financial networks, was presented in [99], underscoring the importance of isolated clique enumeration for network analysis. Isolation concepts have been also explored in temporal networks [163, 164], and applied to practical scenarios such as web structure mining [95], hierarchical representations of scale-free networks [96], and network contraction via isolated cliques [97]. Alternative isolation models have also been proposed, including min- ℓ -isolation [100], τ -isolation based on coreness [165], as well as variants for stars [166], bicliques [167], pseudo-cliques [161, 162], and even approaches inspired by quantum algorithms [168].

Chapter contributions. In this chapter¹, we first show that, when ℓ is part of the input, finding an ℓ -isolated maximal clique is NP-complete. This implies that, unless $P = NP$, there cannot be an output-sensitive polynomial enumeration algorithm, i.e., an algorithm whose running time is polynomial in the number of ℓ -isolated maximal cliques. This motivates considering backtracking-based approaches, rather than reverse-search or proximity-based techniques, which are standard ingredients of output-sensitive and polynomial-delay enumeration algorithms (see, for example, [170, 171]). Therefore, our approach builds upon the well-known Bron-Kerbosch algorithm for maximal clique enumeration [6], improved by a pivoting strategy introduced by Tomita *et al.* [10, 24] (see [Algorithm 1.1](#)). More precisely, we design four pruning heuristics aimed at reducing the search space, proving their correctness. These heuristics, that can be applied individually or in combination, allow the enumeration algorithm to focus only on promising parts of the search tree, discarding branches that are guaranteed not to yield ℓ -isolated maximal cliques. Following a theoretical characterization of the heuristics' pruning power and computational cost, which turn out to be inversely proportional, we conduct an extensive experimental study assessing their practical impact. Our analysis addresses both real and synthetic instances and contributes along several dimensions:

¹Some of the results in this chapter appeared in [169] and are based on joint work with Irene Finocchi and Maurizio Patrignani. We are grateful to Andrea Marino for pointing out the research directions that led to the results in [Section 4.3](#).

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

- It sheds light on the properties of ℓ -isolated cliques relative to the full set of maximal cliques in real-world graphs. This can be particularly useful for guiding the choice of parameter ℓ .
- It gives insights into the relative power of our pruning strategies and of their combinations, identifying the most effective ones.
- It compares our approach with a carefully tuned baseline algorithm [10, 24] and with a top-down state-of-the-art approach to the problem [99]. For avg- ℓ -isolation, two of our variants offer substantial efficiency improvements over the state of the art, by as much as a factor of 4, especially when used on real-world graphs that exhibit social network properties. For max- ℓ -isolation, our algorithms consistently outperform state-of-the-art approaches across all instances and parameter ranges.

Chapter organization. The rest of this chapter is organized as follows. We provide preliminary definitions in Section 4.2. In Section 4.3 we prove that finding an ℓ -isolated maximal clique is NP-complete. The pruning strategies are introduced and theoretically analyzed in Section 4.4. Our experimental setup and the main outcomes of our analysis are discussed in Section 4.5 and Section 4.6, respectively.

4.2 Preliminaries

We refer to Chapter 1 for basic definitions on graphs. Let $G = (V, E)$ be a simple undirected graph, and let n denote the number of vertices of G . We recall that, for any vertex subset $I \subseteq V$, $G[I]$ denotes the subgraph of G induced by I . For a vertex $v \in I$, let $\delta_e(v, I)$ denote its *external degree with respect to I* , i.e., $\delta_e(v, I) = |\{(u, v) \in E \mid u \notin I\}|$. The *external degree of I* , denoted as $\delta_e(I)$, is then the number of edges with one endpoint in I and the other in $V \setminus I$, i.e.,

$$\delta_e(I) = \sum_{v \in I} \delta_e(v, I) = |\{(u, v) \in E \mid v \in I \text{ and } u \in V \setminus I\}|$$

Moreover, given $I, L \subseteq V$ with $I \cap L = \emptyset$, the *external degree of I neglecting L* , denoted as $\delta_e(I, L)$, is the number of edges with one endpoint in I and the other in $V \setminus (I \cup L)$.

The concept of ℓ -isolation has been first introduced in [162]: for any integer value $\ell > 0$, a set $I \subseteq V$ of size $|I| = k$ is *avg- ℓ -isolated*, in G if it has less than $\ell \cdot k$ outgoing edges, i.e., $\delta_e(I) < \ell \cdot k$ (see also Section 3.3.4). For example,

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

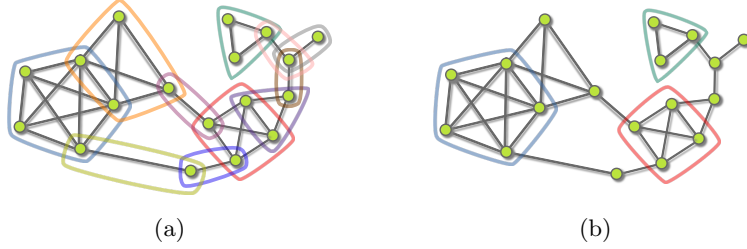


Figure 4.1: (a) An example graph with all its maximal cliques highlighted. (b) Only the 2-isolated maximal cliques are highlighted. In particular, the K_5 maximal clique is avg-2-isolated but not max-2-isolated; the K_4 and the K_3 maximal cliques are both avg-2-isolated and max-2-isolated.

the K_5 maximal clique of [Figure 4.1b](#) is avg-2-isolated because it has $5 < 10$ outgoing edges. We notice that I may contain vertices with more than ℓ external edges: the property of avg- ℓ -isolation holds on average over the vertices of I , implying that the average external degree of vertices in I is less than ℓ . For example, some vertices of the avg-2-isolated K_5 maximal clique of [Figure 4.1b](#) have $\delta_e(v, I) = 2 > \ell - 1$ outgoing edges. Moreover, I is *max- ℓ -isolated* in G if every $v \in I$ has less than ℓ outgoing edges, i.e., $\delta_e(v, I) < \ell$. For example, the K_4 and the K_3 maximal clique of [Figure 4.1b](#) are both max-2-isolated. In this chapter we focus on maximal cliques that are also avg- ℓ -isolated (avg- ℓ -isolated maximal cliques) or max- ℓ -isolated (max- ℓ -isolated maximal cliques).

We remark that any max- ℓ -isolated maximal clique is also avg- ℓ -isolated. Moreover, the property of ℓ -isolation is monotone. Namely, if $\ell_1 > \ell_2 > 0$ then, for both the average and the maximum variants, any ℓ_2 -isolated set is also ℓ_1 -isolated. This follows immediately from the definitions of ℓ -isolation. In particular, for the average case, for an ℓ_2 -isolated set I we have $\delta_e(I) < \ell_2 \cdot k < \ell_1 \cdot k$ when $\ell_1 > \ell_2$. Hence, when searching for ℓ -isolated maximal cliques, the larger is the value of ℓ , the weaker is the isolation of the cliques relative to the rest of the graph, and the larger is the number of ℓ -isolated cliques in the graph. If ℓ is high enough, the set of ℓ -isolated maximal cliques coincides with the set of all maximal cliques of G .

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

4.3 Finding an ℓ -isolated Maximal Clique is NP-complete

In this section we show that, given a graph G and a value of ℓ , finding a single avg- or max- ℓ -isolated maximal clique of G is NP-complete. This implies that, unless $P = NP$, there cannot be an output-sensitive polynomial enumeration algorithm for avg- or max- ℓ -isolated maximal cliques, i.e., an algorithm whose running time is polynomial in the number of solutions found. More formally, we consider the following AVG- ℓ -ISOLATION problem (MAX- ℓ -ISOLATION can be defined analogously):

Problem: AVG- ℓ -ISOLATION.

Input: A graph G and a value $\ell > 0$.

Output: An avg- ℓ -isolated maximal clique of G .

In order to prove that finding an avg- ℓ -isolated maximal clique is NP-hard, we reduce the NP-hard problem 3SAT to the problem of finding an avg- ℓ -isolated maximal clique for a suitable ℓ . Recall that an instance of 3SAT consists of a set C of $c = |C|$ clauses over a set X of variables, where each clause consists of exactly three literals, each being either a variable in X or its negation. 3SAT asks to find a truth assignment for the variables in X such that each clause in C has at least one **true** literal.

Let I_{3SAT} be any instance of 3SAT. We construct an equivalent instance $\langle G, \ell \rangle$ of AVG- ℓ -ISOLATION as follows (refer to [Figure 4.2](#)). For each clause $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ of C , with $i = 1, 2, \dots, c$, we introduce three vertices $v_{i,1}, v_{i,2}, v_{i,3}$ in G . We add an edge between every pair of vertices $(v_{i,j}, v_{h,k})$, with $i \neq h$, such that $l_{i,j}$ is not the negated literal of $l_{h,k}$. Let Δ be the maximum degree of a vertex. We now modify G in such a way that each of its vertices has degree either 1 or Δ . For every vertex $v_{i,j}$ of G with degree $\delta(v_{i,j}) < \Delta$, we add to G vertices $v_{i,j,1}, \dots, v_{i,j,\Delta-\delta(v_{i,j})}$ and we link them to $v_{i,j}$ by means of edges $(v_{i,j}, v_{i,j,1}), \dots, (v_{i,j}, v_{i,j,\Delta-\delta(v_{i,j})})$. Finally, we set $\ell = \Delta - \frac{c-1}{2}$. It is not difficult to see that instance $\langle G, \ell \rangle$ can be constructed in polynomial time in the size of the original instance I_{3SAT} .

Lemma 4.1. *Instance $\langle G, \ell \rangle$ is a positive instance of AVG- ℓ -ISOLATION if and only if I_{3SAT} is a positive instance of 3SAT.*

Proof. Suppose that G admits an avg- ℓ -isolated maximal clique K . By the definition of avg- ℓ -isolation, the number $\delta_e(K)$ of outgoing edges of K is less than $|K| \cdot \ell$. Since each of the $|K|$ vertices of K has degree Δ and $\ell = \Delta - \frac{c-1}{2}$, the number of internal edges of K is at least $|K| \cdot \Delta - |K| \cdot \ell = |K| \cdot (\Delta - \ell) = |K| \cdot \frac{c-1}{2}$. Since K is a clique and has $|K| \cdot \frac{|K|-1}{2}$ edges, we have that $|K| = c$, i.e., K is a

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

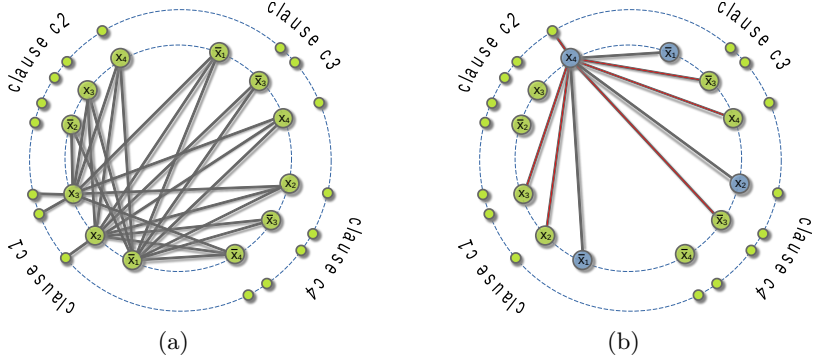


Figure 4.2: The construction of graph G for the 3SAT instance $I_{3\text{SAT}} = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$. (a) This picture only shows the edges incident on the vertices associated with the literals of the first clause. All the vertices have either degree 9 or 1. (b) Filled blue are the vertices corresponding to four literals, one for each clause, that are **true** according to the truth assignment $x_1 = \text{false}$, $x_2 = \text{true}$, $x_3 = \text{false}$, $x_4 = \text{true}$. Only the edges incident to one of the literal are represented (in red the outer edges and in black the internal edges).

clique of size c . Therefore K identifies c literals such that there is an edge in G between any pair of them. This implies that setting these literals to **true** satisfies each of the c clauses of $I_{3\text{SAT}}$ without using two literals that are one the negated of the other, i.e., $I_{3\text{SAT}}$ is satisfiable. Now suppose that $I_{3\text{SAT}}$ is satisfiable and let $l_{1,i_1}, l_{2,i_2}, \dots, l_{c,i_c}$ be a set of true literals, one for each clause $c_1, c_2, \dots, c_c$. We set $K = \{v_{1,i_1}, v_{2,i_2}, \dots, v_{c,i_c}\}$. Observe that the size of K is $|K| = c$. Observe that the number $E(K)$ of edges incident to the vertices of K is at most $E(K) \leq \Delta \cdot c$. Since the internal edges of K are exactly $\frac{c(c-1)}{2}$, the number $\delta_e(K)$ of outgoing edges of K is at most $c \cdot \ell$, because

$$\delta_e(K) = E(K) - \frac{c(c-1)}{2} \leq \Delta \cdot c - \frac{c(c-1)}{2} = c \cdot \left(\Delta - \frac{c-1}{2} \right) = c \cdot \ell$$

Therefore, K is an avg- ℓ -isolated maximal clique of G . $\square$

Since verifying whether a set of vertices is an avg- ℓ -isolated maximal clique is polynomial, problem AVG- ℓ -ISOLATION is trivially in NP. Since the above described construction can be performed in polynomial time and because of [Lemma 4.1](#), we have the following.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

Theorem 4.2. *Problem AVG- ℓ -ISOLATION is NP-complete.*

To prove that finding one max- ℓ -isolated maximal clique is NP-hard we use the same construction described above, with the difference that we set $\ell = \Delta - c + 1$. We claim that G admits a max- ℓ -isolated maximal clique if and only if instance $I_{3\text{SAT}}$ admits a solution. Indeed, suppose $\langle G, \ell \rangle$ admits a max- ℓ -isolated maximal clique K . Each vertex v of K has degree Δ . By the definition of max- ℓ -isolation, v has less than $\ell = \Delta - c + 1$ outgoing edges. Therefore, v has at least $\Delta - (\Delta - c + 1) = c - 1$ internal edges. It follows that K has size $|K| = c$, yielding a solution of $I_{3\text{SAT}}$ as described above. Conversely, suppose that instance $I_{3\text{SAT}}$ admits a solution. We construct a maximal clique K by selecting the vertices of G corresponding to the true literals of the c clauses as in the case of avg- ℓ -isolation. It is immediate to see that K , having size $|K| = c$ and having all vertices of degree Δ , is max- ℓ -isolated with $\ell = \Delta - c + 1$. With analogous reasoning as in the proof of [Theorem 4.2](#) we have the following.

Theorem 4.3. *Problem MAX- ℓ -ISOLATION is NP-complete.*

4.4 Pruning for ℓ -Isolated Maximal Clique Enumeration

In this section we show how to adapt [Algorithm 1.1](#) to enumerate only ℓ -isolated maximal cliques, introducing different pruning strategies to cut uninteresting branches of the search tree. We also analyze the pruning capabilities and computational cost of each strategy, showing an inverse relation that suggests a natural order in which the heuristics may be applied in combination.

We recall from [Chapter 1](#) that in [Algorithm 1.1](#), during the enumeration, C and P are the current candidate clique and the perspective set, respectively. Let T be any maximal clique in the subgraph $G[P]$ induced by P . Thanks to the invariant property that P is always fully connected to C during the execution, it is not difficult to see that $C \cup T$ is a clique in G , though not necessarily ℓ -isolated. Our approach, when processing a node r of the search tree $\mathcal{T}_G$ associated with a triple (C, P, X) , is to avoid recursive calls whenever the subtree rooted at r cannot yield ℓ -isolated maximal cliques.

4.4.1 Pruning for Avg- ℓ -isolation

In this section we consider the enumeration of avg- ℓ -isolated maximal cliques. First, we introduce a necessary condition for a clique grown from C and P to be avg- ℓ -isolated. Let T be any maximal clique in $G[P]$ and let t be its size.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

By the invariant property maintained by [Algorithm 1.1](#), $C \cup T$ is a clique in G . Such a clique has at least $\delta_e(C, P) + |C|(|P| - t)$ external edges: each of the $|P| - t$ vertices not included in $C \cup T$ is indeed connected through an external edge with all the vertices in C . By definition, the number of external edges in an avg- ℓ -isolated clique must be strictly less than ℓ times the clique size. Hence, if

$$\delta_e(C, P) + |C|(|P| - t) \geq \ell(|C| + t) \quad (4.1)$$

then $C \cup T$ is not avg- ℓ -isolated. We now introduce a preliminary result that will be useful to prove the correctness of our pruning strategies.

Lemma 4.4. *Let r be a node in $\mathcal{T}_G$ associated to a candidate set C and a perspective set P . Let P_1 and P_2 be any two subsets of P such that $|P_1| \leq |P_2|$. If $\delta_e(C, P) + |C|(|P| - |P_2|) \geq \ell(|C| + |P_2|)$, then both $C \cup P_2$ and $C \cup P_1$ are not avg- ℓ -isolated.*

Proof. As observed when we introduced [Equation \(4.1\)](#), if $\delta_e(C, P) + |C|(|P| - |P_2|) \geq \ell(|C| + |P_2|)$, then $C \cup P_2$ is not avg- ℓ -isolated. The relationship $|P_1| \leq |P_2|$ yields the following chain of inequalities: $\delta_e(C, P) + |C|(|P| - |P_1|) \geq \delta_e(C, P) + |C|(|P| - |P_2|) \geq \ell(|C| + |P_2|) \geq \ell(|C| + |P_1|)$. Hence, $C \cup P_1$ is also not avg- ℓ -isolated. $\square$

The size of the largest clique in the subgraph $G[P]$ induced by P , i.e., its clique number $\omega(G[P])$, can be used to prune subtrees of $\mathcal{T}_G$. Namely, we prove in [Lemma 4.5](#) that it is correct to prune the subtree rooted at a node labelled (C, P, X) whenever

$$\delta_e(C, P) + |C||P| - \ell|C| \geq \omega(G[P]) \cdot (\ell + |C|) \quad (4.2)$$

Lemma 4.5. *Let r be a node in $\mathcal{T}_G$ associated to a candidate set C and a perspective set P . If $\delta_e(C, P) + |C||P| - \ell|C| \geq \omega(G[P]) \cdot (\ell + |C|)$, then the subtree of $\mathcal{T}_G$ rooted at r contains no leaf corresponding to an avg- ℓ -isolated maximal clique.*

Proof. We use [Lemma 4.4](#), choosing P_2 as a maximum clique in $G[P]$, so that $|P_2| = \omega(G[P])$. Notice that the inequality $\delta_e(C, P) + |C||P| - \ell|C| \geq \omega(G[P]) \cdot (\ell + |C|)$ in the statement can be rewritten as $\delta_e(C, P) + |C|(|P| - \omega(G[P])) \geq \ell(|C| + \omega(G[P]))$. When it is satisfied, by [Equation \(4.1\)](#) the subgraph induced by C extended by a maximum clique of $G[P]$ cannot be avg- ℓ -isolated. By [Lemma 4.4](#), if we extend C by any other clique of $G[P]$, the resulting subgraph is also not avg- ℓ -isolated. Any clique P_1 of $G[P]$ is indeed

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

smaller than or equal to a maximum clique, i.e., $|P_1| \leq |P_2| = \omega(G[P])$ and thus [Lemma 4.4](#) can be applied. $\square$

[Lemma 4.5](#) implies that a check on $\omega(G[P])$ would be sufficient to prune at node r . Unfortunately, computing $\omega(G[P])$ is difficult — max-clique being NP-hard. Hence, we exploit upper bounds to $\omega(G[P])$ to guide pruning, inspired by the approach suggested in [\[87\]](#) for the computation of τ -visible summaries. We will prove in [Theorem 4.6](#) that this maintains correctness with respect to our avg- ℓ -isolation problem. In particular, $\omega(G[P])$ cannot be larger than:

1. the number $|P|$ of vertices in $G[P]$;
2. the maximum degree of $G[P]$, denoted by $\Delta(G[P])$, augmented by 1;
3. the maximum value of k such that $G[P]$ contains at least k vertices with degree $\geq k - 1$, denoted by $\tau(G[P])$;
4. the degeneracy of $G[P]$, denoted by $\kappa(G[P])$, augmented by 1. The degeneracy is the largest value of k for which $G[P]$ has a k -core (i.e., a maximal subgraph in which each vertex has degree at least k).

The first upper bound $\omega(G[P]) \leq |P|$ is the weakest, while the fourth one $\omega(G[P]) \leq \kappa(G[P])$ is the strictest. In particular, the following chain of inequalities holds:

$$\omega(G[P]) \leq \kappa(G[P]) + 1 \leq \tau(G[P]) \leq \Delta(G[P]) + 1 \leq |P| \quad (4.3)$$

This can be intuitively explained as follows. The fact that $\omega(G[P]) \leq \kappa(G[P]) + 1$ is well-known in the literature: in every clique of size t , every vertex has indeed degree $t - 1$ and is thus part of a $(t - 1)$ -core, proving that the degeneracy is at least $t - 1$. The value $\tau(G[P])$ represents a relaxed version of degeneracy, and thus $\kappa(G[P]) + 1 \leq \tau(G[P])$. Indeed, the degeneracy $\kappa(G[P])$ can be computed by repeatedly removing a vertex of minimum degree from the graph, and updating the degree of remaining vertices along the way. Its relaxation $\tau(G[P])$ can be instead obtained by removing vertices based on their original degrees, without updates. Hence, it may be the case that more vertices satisfy the degree condition at point 3. Consider, for instance, a complete binary tree on four levels: $\tau(G[P]) = 4$, as the maximum degree 3 is attained by all the 6 vertices in the two internal layers, while $\kappa(G[P]) + 1 = 2$, as the degeneracy of a tree is 1. Finally, it can be easily seen that $\tau(G[P]) \leq \Delta(G[P]) + 1$, because we cannot have k vertices of degree larger than or equal to $k - 1$ if $k \geq \Delta(G[P]) + 2$.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

We use the upper bounds on $\omega(G[P])$ to obtain lower bounds on the total number of external edges that any maximal clique T , grown from the current clique C and the candidate set P , can have. Namely, our algorithm prunes the computation at node (C, P, X) whenever

$$\delta_e(C, P) + |C||P| - \ell|C| \geq \bar{\omega}(G[P]) \cdot (\ell + |C|) \quad (4.4)$$

where $\bar{\omega}(G[P])$ can be any of the quantities involved in [Equation \(4.3\)](#) (i.e., either $\omega(G[P])$ or any of its upper bounds). In the following we discuss the correctness of the pruning condition expressed by [Equation \(4.4\)](#), proving that it is safe to prune using upper bounds $\bar{\omega}$ on ω , instead of ω itself.

Theorem 4.6. *Given a graph G and a positive integer ℓ , let $\mathcal{T}_G$ be the search tree of [Algorithm 1.1](#) executed on G , let r be a node in $\mathcal{T}_G$ associated to a candidate set C and a perspective set P , and let $\bar{\omega}(G[P])$ be any upper bound on $\omega(G[P])$. If $\delta_e(C, P) + |C||P| - \ell|C| \geq \bar{\omega}(G[P]) \cdot (\ell + |C|)$, then the subtree of $\mathcal{T}_G$ rooted at r contains no leaf corresponding to an avg- ℓ -isolated maximal clique.*

Proof. We exploit [Lemma 4.4](#), choosing P_1 as a maximum clique in $G[P]$ and choosing P_2 as follows: if the upper bound $\bar{\omega}(G[P])$ in [Equation \(4.3\)](#) is $|P|$, then P_2 is P itself; if instead it is $\Delta(G[P]) + 1$, then P_2 is formed by any vertex of maximum degree in P together with its neighbors; if it is $\tau(G[P])$, then P_2 is formed by any $\tau(G[P])$ vertices of P each having degree at least $\tau(G[P]) - 1$; and when it is $\kappa(G[P]) + 1$, then P_2 is the $\kappa(G[P])$ -core of P . Hence, $|P_1| = \omega(G[P]) \leq \bar{\omega}(G[P]) = |P_2|$. If the test in [Equation \(4.4\)](#) is verified, by [Lemma 4.4](#) we have that the subgraph induced by C extended by a maximum clique of $G[P]$ is not avg- ℓ -isolated. In particular, from the proof of [Lemma 4.4](#) we have $\delta_e(C, P) + |C|(|P| - \omega(G[P])) \geq \ell(|C| + \omega(G[P]))$ ([Equation \(4.2\)](#)). This in turn implies, by [Lemma 4.5](#), that any other clique in $G[P]$ is not avg- ℓ -isolated and that pruning at r does not entail discarding any avg- ℓ -isolated clique, thereby proving the statement. $\square$

We conclude with a remark on the running time: the quantities $|P|$, $\Delta(G[P])$, $\tau(G[P])$, and $\kappa(G[P])$ involved in [Equation \(4.3\)](#) can be computed in $\mathcal{O}(1)$, $\mathcal{O}(1)$, $\mathcal{O}(|P|)$, and $\mathcal{O}(|E(G[P])|)$ time, respectively, yielding an inverse relation between running time and pruning power.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

4.4.2 Pruning for Max- ℓ -isolation

We now consider the enumeration of max- ℓ -isolated maximal cliques, extending the theoretical framework introduced for avg- ℓ -isolation in [Section 4.4.1](#). Given a candidate set C and a perspective set P , we first present a necessary condition for a clique grown from C and P to be max- ℓ -isolated.

Lemma 4.7. *Let r be a node in $\mathcal{T}_G$ associated to a candidate set C and a perspective set P . Let t be the size of a maximal clique $T \subseteq P$. If there exists a vertex $v \in C$ such that*

$$\delta(v) - |C| + 1 - t \geq \ell \quad (4.5)$$

then $C \cup T$ is not a max- ℓ -isolated maximal clique.

Proof. Consider the clique $C \cup T$ and vertex $v \in C$ that satisfies [Equation \(4.5\)](#). Since each of the $|P| - t$ vertices not included in $C \cup T$ is connected with v , v has $\delta_e(v, C \cup P) + |P| - t$ edges external to $C \cup T$. However, from [Equation \(4.5\)](#) it can be obtained

$$\delta_e(v, C \cup P) + |P| - t \geq \ell \quad (4.6)$$

which proves that $C \cup T$ is not max- ℓ -isolated. Indeed, by the invariant property maintained by [Algorithm 1.1](#), v is connected to the other $|C| - 1$ vertices of C and to all the vertices of P . This implies that $\delta(v) = \delta_e(v, C \cup P) + |C| - 1 + |P|$, which, together with [Equation \(4.5\)](#) yields [Equation \(4.6\)](#). $\square$

Similarly to avg- ℓ -isolation and using the notation introduced in [Section 4.4.1](#), we can now prove the following theorem:

Theorem 4.8. *Given a graph G and a positive integer ℓ , let $\mathcal{T}_G$ be the search tree of [Algorithm 1.1](#) executed on G , and let r be a node in $\mathcal{T}_G$ associated to a candidate set C and a perspective set P . Denote by $\omega(G[P])$ the size of the maximum clique in $G[P]$ and by d the maximum degree in G of a vertex belonging to C . Given any upper bound $\bar{\omega}(G[P])$ on $\omega(G[P])$, if $d - |C| + 1 - \ell \geq \bar{\omega}(G[P])$, then the subtree of $\mathcal{T}_G$ rooted at r contains no leaf corresponding to a max- ℓ -isolated maximal clique.*

Proof. If $d - |C| + 1 - \ell \geq \bar{\omega}(G[P])$, then by definition of $\bar{\omega}(G[P])$, we have that $d - |C| + 1 - \ell \geq \omega(G[P])$. This implies that $d - |C| + 1 - \ell \geq t$, for every maximal clique T of $G[P]$ of size t . Hence, the vertex of C of maximum degree d satisfies [Equation \(4.5\)](#) and [Lemma 4.7](#) guarantees that $C \cup T$ is not max- ℓ -isolated. Thus, the subtree of $\mathcal{T}_G$ rooted at r contains no leaf corresponding to a max- ℓ -isolated maximal clique. $\square$

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

Table 4.1: Statistics of the real-world benchmarks. Δ is the maximum degree, and d is the degeneracy.

Graph	ID	Category	Vertices	Edges	Max. Cliques	Δ	d
brightkite	bk	social	58,228	214,078	290,004	1,134	52
livemocha	lm	social	104,103	2,193,083	3,711,569	2,980	92
gowalla	gw	social	196,591	950,327	1,212,679	14,730	51
twitter	ws	social	465,017	833,540	805,170	677	30
youtube	cy	social	1,134,890	2,987,624	3,265,956	28,754	51
hyves	hy	social	1,402,673	2,777,419	2,590,217	31,883	39
ca-astroph	ap	co-authorship	18,771	198,050	36,427	504	56
dblp	cd	co-authorship	317,080	1,049,866	257,551	343	113
cfinder-google	gc	hyperlink	15,763	148,585	75,258	11,401	102
stanford	sf	hyperlink	281,903	1,992,636	1,055,936	38,625	71
italian-cnr	ic	hyperlink	325,557	2,738,969	1,425,378	18,236	83
notre-dame	nd	hyperlink	325,729	1,090,108	495,947	10,721	155
baidu	bar	hyperlink	415,624	2,374,044	4,596,143	127,066	228
web-google	go	hyperlink	875,713	4,322,051	1,417,580	6,332	44
cit-hepph	phc	citation	34,546	420,877	412,493	846	30
citeseer	cs	citation	384,054	1,736,145	1,232,951	1,739	15

Based on [Theorem 4.8](#), an efficient algorithm for the computation of max- ℓ -isolated maximal cliques can be devised by pruning the computation at a node (C, P, X) of $\mathcal{T}_G$ whenever $d - |C| + 1 - \ell \geq \bar{\omega}(G[P])$, where $d = \max_{v \in C} \delta(v)$ and $\bar{\omega}(G[P])$ is any of the upper bounds on $\omega(G[P])$ discussed in [Section 4.4.1](#) (see [Equation \(4.3\)](#)).

4.5 Experimental Setup

Algorithms under evaluation. The four different heuristics discussed in [Section 4.4](#) are denoted in our analysis by **size**, **degree**, **softcore**, and **degeneracy**, respectively. We also consider a combined approach, called **combo**, that orderly integrates the **size** and **softcore** heuristics: we tested many different combinations of the four basic heuristics, and **combo** consistently proved to be the most effective one. For the sake of comparison, we include the algorithm proposed in [\[99\]](#), denoted by **HKMN**. We relied on the implementation used in [\[99\]](#). Finally, we consider a baseline algorithm, derived from [Algorithm 1.1](#) [\[10, 24\]](#) and denoted by **TTT**, that filters each non- ℓ -isolated maximal clique instead of reporting it.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

Benchmarks. We conducted experiments on a variety of real-world and synthetic graphs. Real-world graphs, which are the main target of our analysis, are taken primarily from the KONECT repository (<http://konect.cc>) [73]. Throughout this chapter we focus on a representative subset of 16 graphs, whose main properties are summarized in Table 4.1. Our selection was guided by diversity criteria, taking into account *graph size*, *number of maximal cliques*, *maximum degree* Δ , and *degeneracy* d . To further ensure heterogeneity, our benchmarks include different types of networks, namely social, hyperlink, co-authorship, and citation graphs. For the sake of completeness, the number of ℓ -isolated maximal cliques for different values of ℓ , as well as their percentage with respect to the total number of maximal cliques, is reported for these graphs in Table 4.2.

We also used synthetic instances generated according to two different models:

- Random scale-free networks generated using a preferential attachment mechanism as proposed by Barabási and Albert [172]. $BA_{n,m}$ networks are grown by sequentially adding new nodes, each with m edges, that attach preferentially to existing nodes in proportion to their degrees. We used $n = 100,000$ and $m \in \{25, 50\}$.
- Random networks based on the $G_{n,m,p}$ model (see [173] and the references therein), which is also used in [99]. These networks have n vertices; each of them is assigned m features, each with probability p ; and all the vertices with the same feature form a clique. On these instances, for the sake of reproducibility, we selected parameter values similar to [99], varying n in $[50, 550]$ with step 50, m in $[5, 95]$ with step 10, p in $[0.025, 0.25]$ with step 0.025, and ℓ in $[10, 200]$ with step 10.

Performance metrics. In addition to running times, we profiled the number of recursive calls and contrasted these values with the number of ℓ -isolated cliques. The running time of each heuristic depends indeed both on its pruning capabilities and on the overhead introduced by the pruning test in Equation (4.4): the more complex the pruning strategy is, the larger the test overhead.

Experimental platform. All the experiments were run on a laptop equipped with an Intel® Core™ i5-1135G7 CPU (4 cores, 4.2 GHz) and 8 GB of RAM. Our algorithms were implemented in OCaml (<https://ocaml.org>) to ensure a fair comparison with the available implementation used in [99].

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

Table 4.2: Evolution of the number of ℓ -isolated maximal cliques and the percentage with respect to all maximal cliques for different values of ℓ .

Graph	$\ell = 1$	$\ell = 10$	$\ell = 20$	$\ell = 30$	$\ell = 40$	$\ell = 50$	$\ell = 100$	$\ell = 150$	$\ell = 200$	$\ell = 250$
brightkite	2346 (0.81%)	32661 (11.26%)	54118 (18.66%)	67631 (23.32%)	77577 (26.75%)	85257 (29.40%)	113328 (39.08%)	181361 (62.54%)	232791 (80.27%)	264373 (91.16%)
livemocha	2 (0.00%)	6619 (0.18%)	32211 (0.87%)	69514 (1.87%)	112054 (3.02%)	156226 (4.21%)	361422 (9.74%)	528104 (14.23%)	668495 (18.01%)	792579 (21.35%)
gowalla	3993 (0.33%)	107537 (8.87%)	176808 (14.58%)	220693 (18.20%)	252900 (20.85%)	279548 (23.05%)	376567 (31.05%)	440791 (36.35%)	511168 (42.15%)	565420 (46.63%)
twitter	8 (0.00%)	803 (0.10%)	3401 (0.42%)	6678 (0.83%)	10471 (1.30%)	15811 (1.96%)	47327 (5.88%)	82293 (10.22%)	124235 (15.43%)	603666 (74.97%)
youtube	41563 (1.27%)	549757 (16.83%)	756313 (23.16%)	882061 (27.01%)	975738 (29.88%)	1049952 (32.15%)	1291485 (39.54%)	1450548 (44.41%)	1566338 (47.96%)	1662987 (50.92%)
hyves	44 (0.00%)	829384 (32.02%)	1218221 (47.03%)	1295354 (50.01%)	1394493 (53.84%)	1515859 (58.52%)	1920881 (74.16%)	2018340 (77.92%)	2113061 (81.58%)	2190694 (84.58%)
ca-astroph	384 (1.05%)	2629 (7.22%)	5157 (14.16%)	7462 (20.48%)	9732 (26.72%)	11941 (32.78%)	21594 (59.28%)	28811 (79.09%)	32935 (90.41%)	35077 (96.29%)
dblp	4762 (1.85%)	103815 (40.31%)	168752 (65.52%)	205317 (79.72%)	226139 (87.80%)	237806 (92.33%)	255468 (99.19%)	257404 (99.94%)	257537 (99.99%)	257550 (100.00%)
cfinder-google	92 (0.12%)	657 (0.87%)	1097 (1.46%)	1471 (1.95%)	1597 (2.12%)	1833 (2.44%)	3679 (4.89%)	5623 (7.47%)	6858 (9.11%)	17418 (23.14%)
stanford	1223 (0.12%)	65655 (6.22%)	107570 (10.19%)	133984 (12.69%)	153565 (14.54%)	170323 (16.13%)	217294 (20.58%)	234531 (22.21%)	249591 (23.64%)	262519 (24.86%)
italian-cnr	1455 (0.10%)	89276 (6.26%)	141082 (9.90%)	172788 (12.12%)	193906 (13.60%)	210687 (14.78%)	267492 (18.77%)	320474 (22.48%)	369278 (25.91%)	399686 (28.04%)
notre-dame	1292 (0.26%)	108550 (21.89%)	175070 (35.30%)	211801 (42.71%)	234167 (47.22%)	251205 (50.65%)	290741 (58.62%)	317485 (64.02%)	332918 (67.13%)	345365 (69.64%)
baidu	22792 (0.50%)	173799 (3.78%)	306856 (6.68%)	406504 (8.84%)	466079 (10.14%)	519940 (11.31%)	631133 (13.73%)	1031465 (22.44%)	3280483 (71.37%)	3301681 (71.84%)
web-google	8745 (0.62%)	324503 (22.89%)	535311 (37.76%)	643214 (45.37%)	709705 (50.06%)	757338 (53.42%)	888481 (62.68%)	944254 (66.61%)	975307 (68.80%)	1003754 (70.81%)
cit-hepph	141 (0.03%)	7273 (1.76%)	23354 (5.66%)	45525 (11.04%)	71242 (17.27%)	98142 (23.79%)	251893 (61.07%)	355771 (86.25%)	390060 (94.56%)	402261 (97.52%)
citeseer	10965 (0.89%)	247124 (20.04%)	478583 (38.82%)	635481 (51.54%)	746623 (60.56%)	829906 (67.31%)	1043603 (84.64%)	1126876 (91.40%)	1164634 (94.46%)	1187575 (96.32%)

4.6 Experimental Results

In this section we present the key findings of our experimental analysis, both for avg- and max- ℓ -isolation. [Section 4.6.1](#) examines the properties of ℓ -isolated cliques in real-world benchmarks. We evaluate our pruning strategies in [Section 4.6.2](#), identifying the most effective ones, which are compared against state-of-the-art methods in [Section 4.6.3](#). We complement our study with results on synthetic instances in [Section 4.6.4](#).

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

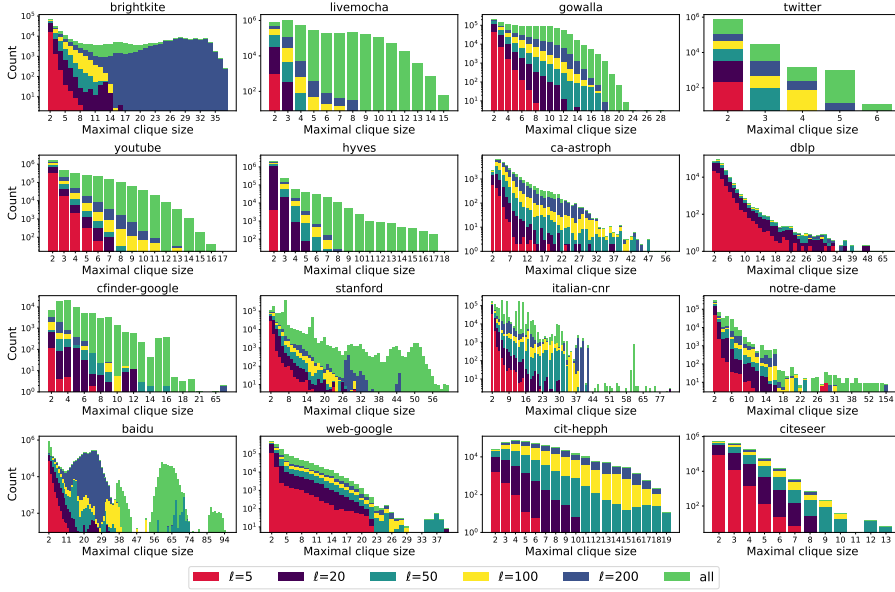


Figure 4.3: Avg- ℓ -isolated maximal clique size distribution for different values of ℓ .

4.6.1 On the Distribution of ℓ -isolated Maximal Cliques

In Figure 4.3 we summarize the distribution of maximal cliques by size, together with their breakdown into avg- ℓ -isolated cliques for different values of ℓ .

The upper profile of the stacked bars shows the total number of maximal cliques with a given number of vertices. In all benchmarks, most cliques consist of a few vertices: the number of cliques grows quickly for small sizes, then drops sharply. The maximum clique size varies substantially across graph types (cfr. Table 4.1), reflecting different connectivity patterns: the first six social networks form only small cliques (typically less than 20 vertices), the two co-authorship networks and the two citation networks reach moderate sizes, while the six Web graphs contain the largest cliques, sometimes exceeding 150 vertices. The difference can be best appreciated on graphs of different types but similar size: compare, e.g., the maximum clique size in *gowalla*, *dblp*, and *notre-dame*, which have comparable numbers of edges.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

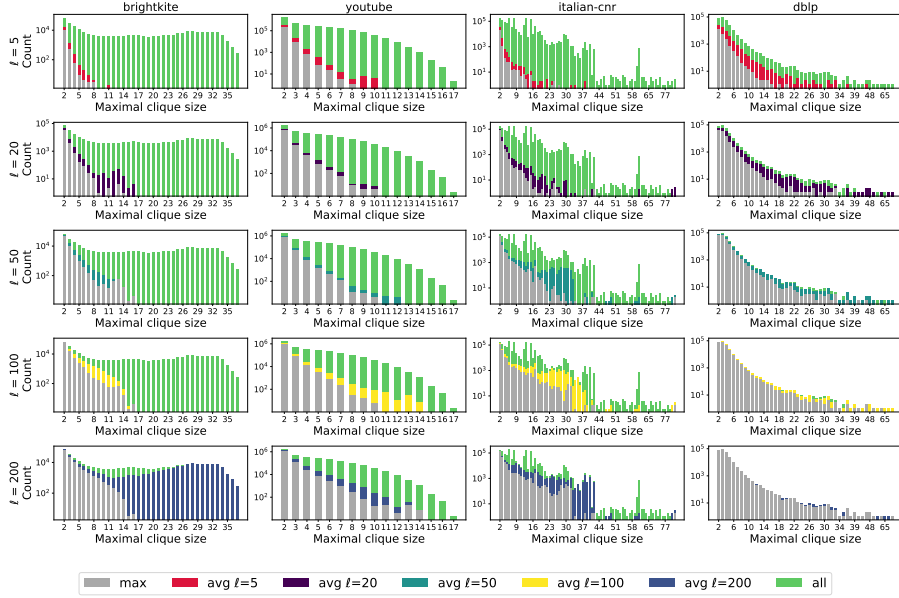


Figure 4.4: Distribution of maximal clique sizes for different values of ℓ , comparing max- ℓ -isolated and avg- ℓ -isolated maximal cliques. Colored bars represent the ℓ -isolated maximal cliques, while the gray bars correspond to the max- ℓ -isolated maximal cliques.

This analysis can be refined by considering the color segments within each bar of Figure 4.3, that show how many maximal cliques are avg- ℓ -isolated for several values of ℓ . For small values of ℓ (e.g., $\ell \leq 20$), the isolation requirement is very strict and only a small fraction of small cliques qualify: large cliques are typically excluded, as they tend to be embedded in denser regions of the graph with many external connections. The distributions for small ℓ are thus sharply truncated and dominated by cliques with just a few vertices. As ℓ increases, the number of avg- ℓ -isolated cliques grows, being the property monotone (see Section 4.2). In addition, larger values of ℓ allow increasingly larger cliques to appear in the summary, extending the tail of the distribution. For large thresholds, such as $\ell = 200$, many more cliques are admitted: in many networks (e.g., brightkite, web-google, cit-hepph) the profile for 200-isolated cliques approaches that of the total distribution, making the avg- ℓ -isolation criterion

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

almost ineffective.

The number of max- ℓ -isolated maximal cliques is upper bounded by the number of avg- ℓ -isolated maximal cliques. This is apparent in Figure 4.4, where the number of max- ℓ -isolated maximal cliques is depicted in gray (independently of ℓ). We observed that, across most of the networks considered in Table 4.1, the distributions of max- ℓ -isolated and avg- ℓ -isolated maximal cliques behave very similarly. Some networks for which the two distributions show some notable differences are represented in Figure 4.4. For example, the network **brightkite** (first column of Figure 4.4) shows a drop of the number of max- ℓ -isolated maximal cliques for $\ell = 200$ (fifth row of Figure 4.4). A similar phenomenon can be observed for **italian-cnr** (third column of Figure 4.4) for values of ℓ greater or equal than 50. Overall, neglecting these few exceptions, the number of max- ℓ -isolated maximal cliques can be considered as a reasonable lower bound approximation of the number of avg- ℓ -isolated maximal cliques, especially since such a number can be more efficiently computed as shown in Section 4.6.3.

4.6.2 Assessment of Different Pruning Strategies

We now assess the effectiveness of different pruning strategies considering two different metrics: number of recursive calls and running time. We first focus on avg- ℓ -isolation. Because the results are largely consistent across different values of ℓ , Figure 4.5 shows only the case $\ell = 50$. The left radar plot in Figure 4.5 reports the percentage of recursive calls with respect to the number one would have without pruning. This gives a quantitative view of the pruning power of the heuristics: the larger the area, the weaker the pruning. In spite of a large variability across benchmarks, **size** is always the weakest, as expected from the theoretical analysis, producing the largest number of calls. On some benchmarks, e.g., **twitter**, this number can be much larger than the other strategies and very close to the baseline. As expected, **combo** and **softcore** overlap perfectly, since **combo** is **size** followed by **softcore** whenever **size** fails, thus inheriting the pruning power of **softcore**. While **degeneracy** consistently achieves the strongest pruning, yielding the smallest areas across all datasets, it is rather similar to **softcore** and **combo**.

Interestingly, the picture is reversed in the right radar plot, which reports the running time normalized with respect to the slowest execution: **degeneracy**, which is the most powerful pruning strategy, is also almost always the slowest, with the exception of **baidu**, where **degree** performs worst. The heuristics with weaker pruning, most notably **size**, **softcore** and **combo**, are instead the fastest. Among these, **softcore** and **combo** behave very similarly, with

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

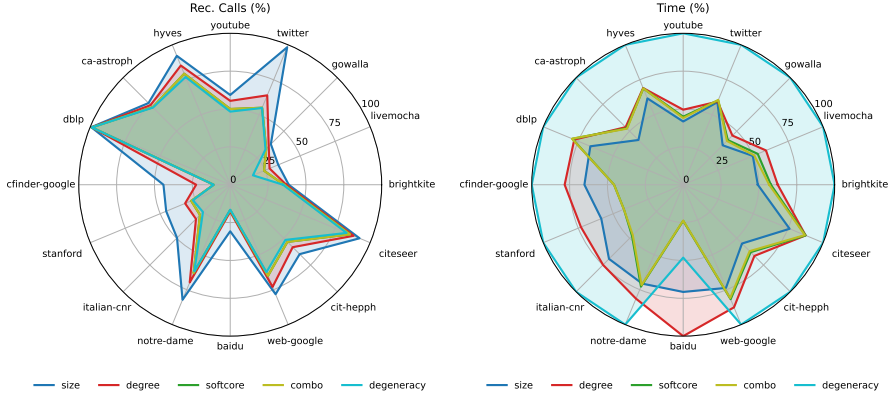


Figure 4.5: Comparison of the recursive calls (%) and the running time (%) with respect to the slowest execution) of different pruning strategies for $\ell = 50$.

combo being slightly preferable in some cases. **degree** exhibits an intermediate performance across benchmarks. Overall, the results highlight a clear trade-off between pruning power and efficiency: the running times observed in theory to perform the pruning tests (see Section 4.4.1) appear to have a stronger impact on performance than the actual amount of pruned nodes.

The result for max- ℓ -isolation confirmed those in the average case, with heuristics **degree**, **softcore**, and **degeneracy** generally outperformed by the others. In the remainder of this section we will thus focus on **size** and **combo**, which appear to be the most effective strategies both for avg- and max- ℓ -isolation.

4.6.3 Experiments on Real-world Networks

In Figure 4.6 we compare our best heuristics for avg- ℓ -isolation, i.e., **size** and **combo**, against HKMN and TTT, also analyzing how ℓ affects the running times. We chose a range of values for ℓ based on the observations in Section 4.6.1, covering both the lower and the upper spectrum of the isolation factor. A few trends emerge clearly from the charts. First, the running time of TTT is independent of ℓ , thus remaining constant. The two heuristics **size** and **combo** typically deliver the best performance over a large range of values of ℓ , with runtimes that remain low and scale smoothly as ℓ increases, when larger portions of $\mathcal{T}_G$ need to be explored. Their curves are often close (see, e.g., **livemocha** or

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

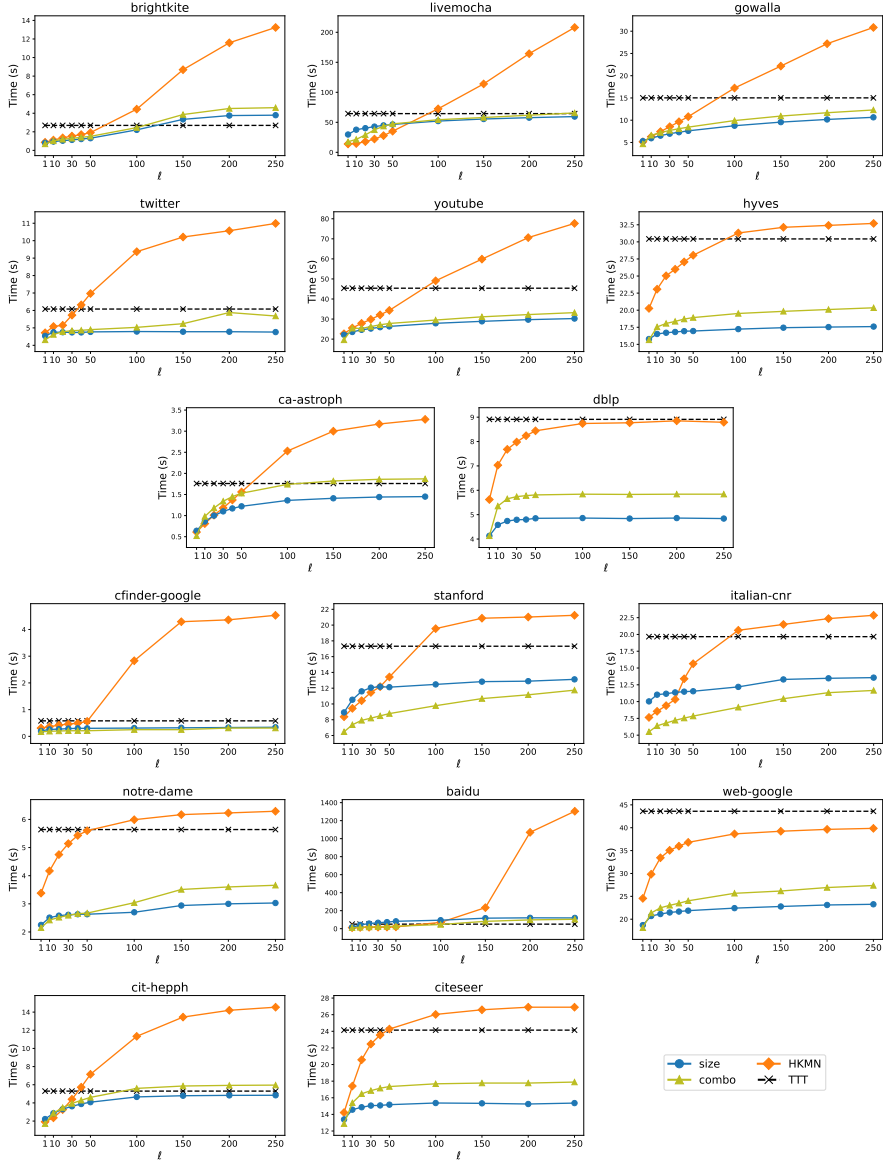


Figure 4.6: Comparison of size, combo, HKMN, and TTT with avg-isolation on real-world graphs.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

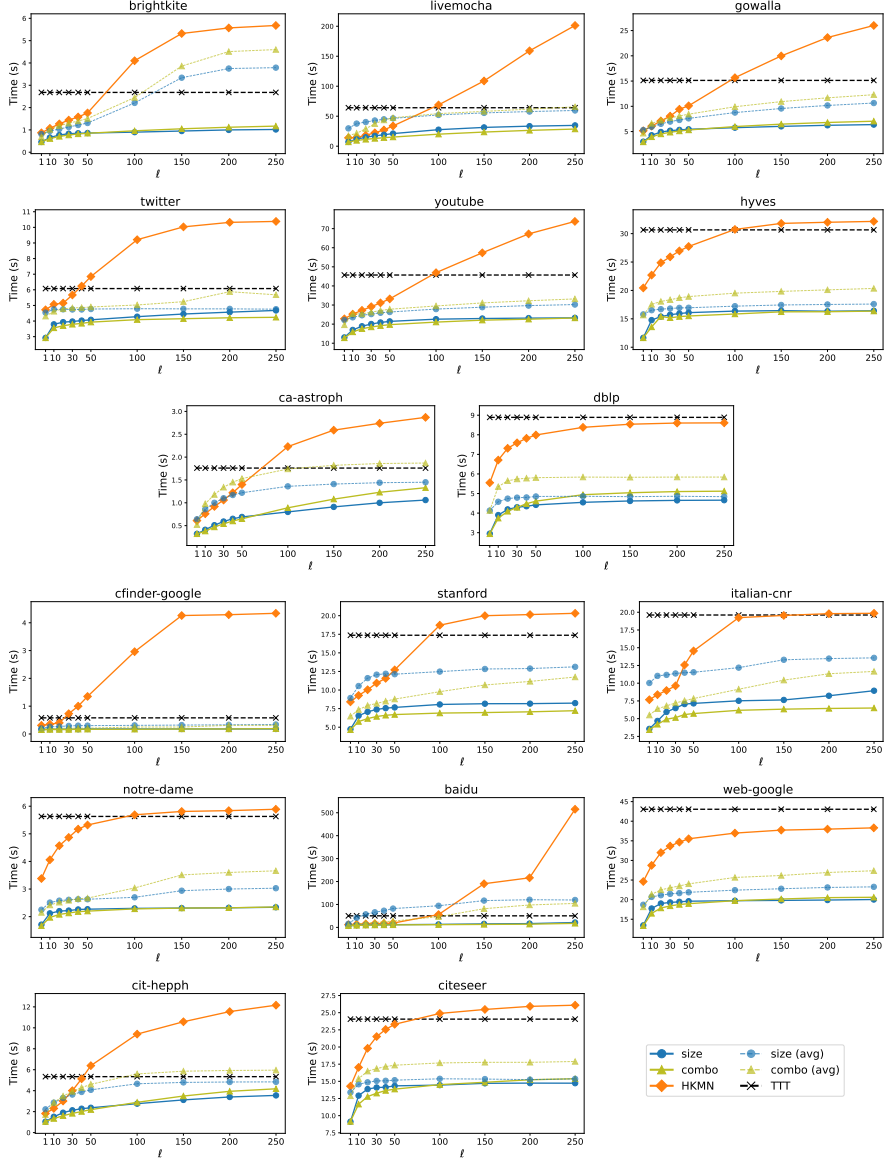


Figure 4.7: Comparison of `size`, `combo`, `HKMN`, and `TTT` with max-isolation on real-world graphs.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

`cfinder-google`), `combo` is marginally faster than `size` on a few instances (most notably, `stanford` and `italian-cnr`), but `size` seems preferable in most of the other ones. On the other side, HKMN is by far the slowest algorithm when ℓ gets large, with runtimes that grow quickly with ℓ : this is due to an ℓ -parameterized subroutine for finding minimal vertex covers. It can be instead competitive when ℓ is very small. TTT is only occasionally competitive with `size` and `combo`, but still outperforms HKMN for large ℓ except on `web-google` and `dblp`. The observed relative differences between the algorithms persist across the datasets. Even weak pruning heuristics that add a small test overhead on nodes of $\mathcal{T}_G$ translate into significant speedups, which can be as large as $4\times$ and $2\times$ with respect to HKMN and TTT, respectively.

As shown in [Figure 4.7](#), max- ℓ -isolation algorithms (solid curves) consistently exhibit lower running times than avg- ℓ -isolation (dashed curves). Also, for all tested values of ℓ , heuristics `size` and `combo` are faster than both the exhaustive search TTT and the heuristic HKMN, clearly emerging as the algorithms of choice in practice.

4.6.4 Experiments with Synthetic Networks

We now report experiments on synthetic instances that mostly confirm the analysis on real-world graphs.

We first discuss experiments on $\text{BA}_{n,m}$ networks. [Figure 4.8](#) compares TTT, HKMN, `size`, and `combo`. For $m = 25$ and $\ell \leq 20$ ([Figure 4.8](#), left plot), we have that HKMN is comparable with `combo` and both of them perform better than `size`. However, the dashed red line also shows that in the above range the ℓ -isolated maximal cliques are very few (we also experimentally assessed their small size, akin to real-world networks). For higher values of ℓ , the running time of HKMN seems to be strongly dependent on the output size (see also [Figure 4.12](#) for the full plot of HKMN). Heuristic `size` (and sometimes `combo`), instead, provide a more efficient solution when $\ell \geq 30$. Curve trends are similar when $m = 50$ ([Figure 4.8](#), right plot). The percentage of ℓ -isolated cliques decreases: for $\ell = 250$ it is barely above 60% and remains close to zero for $\ell \leq 50$ (in this range HKMN is the fastest algorithm). When $\ell > 50$, TTT is faster than `combo` and HKMN, but `size` remains the preferred solution.

Overall, for small values of ℓ , `combo` and `size` can be either slightly slower or slightly faster than HKMN. However, in this regime the output set of avg- ℓ -isolated maximal cliques is empty. Whenever the output set is non-empty, the proposed heuristics are consistently and often substantially faster than HKMN, and in many cases also significantly faster than TTT.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

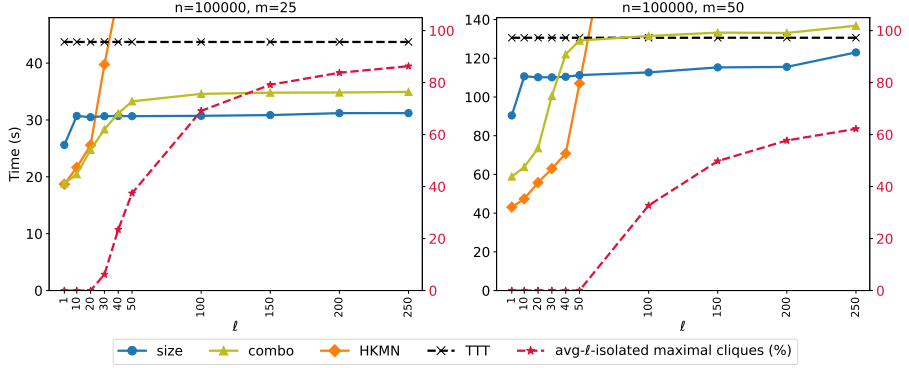


Figure 4.8: Results on $BA_{n,m}$ synthetic graphs for avg- ℓ -isolation. The dashed red line shows the number of avg- ℓ -isolated maximal cliques relative to the total number.

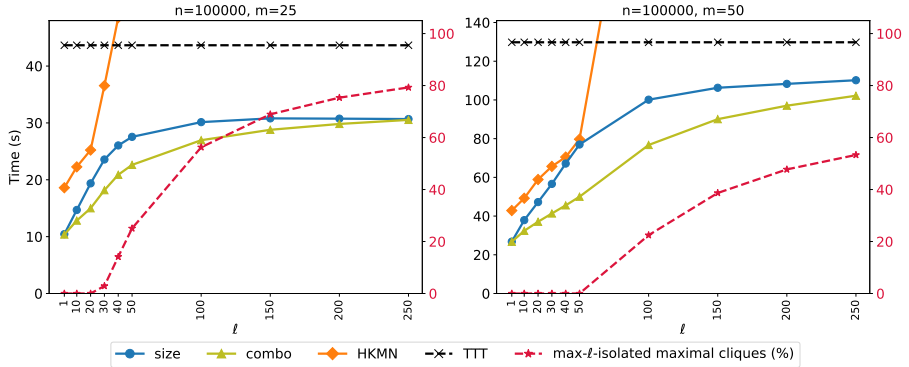


Figure 4.9: Results on $BA_{n,m}$ synthetic graphs for max- ℓ -isolation. The dashed red line shows the number of max- ℓ -isolated maximal cliques relative to the total number.

The analogous experiments for max- ℓ -isolation are even more positive than those for avg- ℓ -isolation (see Figure 4.9) and show that **combo** and **size** outperform both TTT and HKMN, for all values of ℓ (see also Figure 4.13 for the full plot of HKMN).

Networks generated according to the $G_{n,m,p}$ model have substantially differ-

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

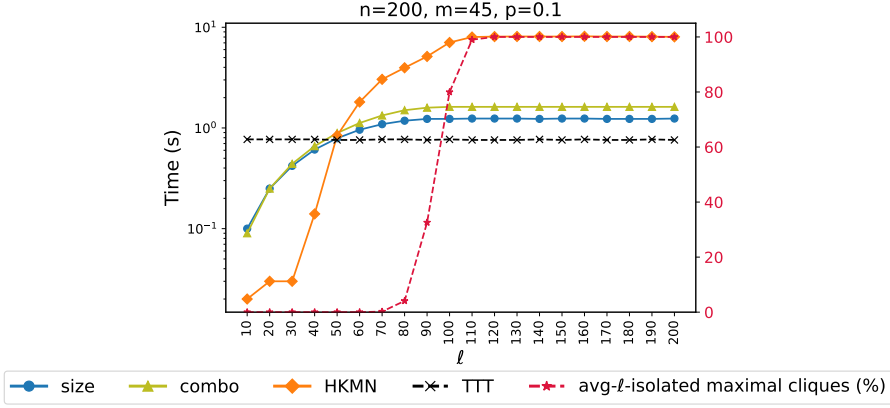


Figure 4.10: Results on $G_{n,m,p}$ synthetic graphs avg- ℓ -isolation. The dashed red line shows the number of avg- ℓ -isolated maximal cliques relative to the total number.

ent properties: they are typically very dense and ℓ -isolated maximal cliques are barely present for small values of ℓ (e.g., $\ell < 80$). Conversely, for high values of ℓ (e.g., $\ell > 100$), their percentage becomes close to 100%, after a very steep transition (see the dashed red line in Figure 4.10). We reproduced the analysis of [99]. In Figure 4.10 we report the results with respect to ℓ that confirm those in [99], with HKMN outperforming other approaches for small values of ℓ . TTT seems instead the algorithm of choice for larger values.

The analogous experiments for max- ℓ -isolation are much more positive for **size** and **combo** (see Figure 4.11). The two heuristics are the fastest approaches for $\ell \leq 110$. Not surprisingly, when the percentage of max- ℓ -isolated maximal cliques approaches 100%, the enumeration strategy of TTT offers an efficient alternative to any other heuristic. In any case, $G_{n,m,p}$ graphs are very far from the properties of real-world networks and appear to represent worst-case scenarios for all the algorithms proposed in this chapter.

For the sake of completeness, we also report the full running time comparison between HKMN and TTT on $BA_{n,m}$ synthetic graphs, considering both avg- and max- ℓ -isolation. Figure 4.12 and Figure 4.13 complement Figure 4.8 and Figure 4.9, respectively.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

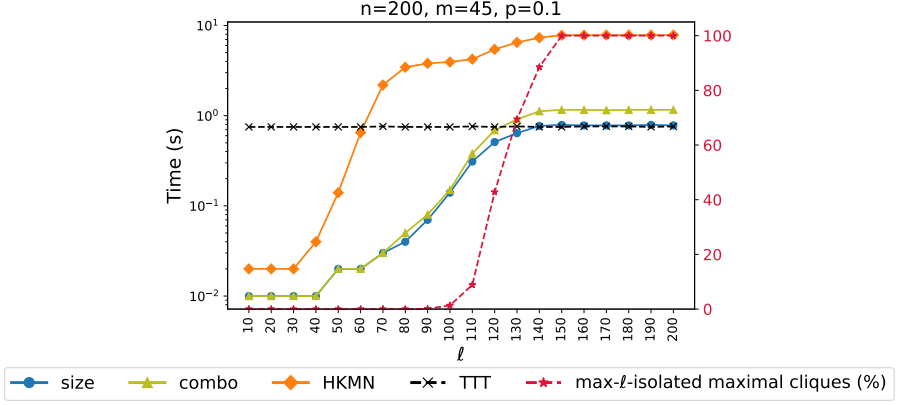


Figure 4.11: Results on $G_{n,m,p}$ synthetic graphs for max- ℓ -isolation. The dashed red line shows the number of max- ℓ -isolated maximal cliques relative to the total number.

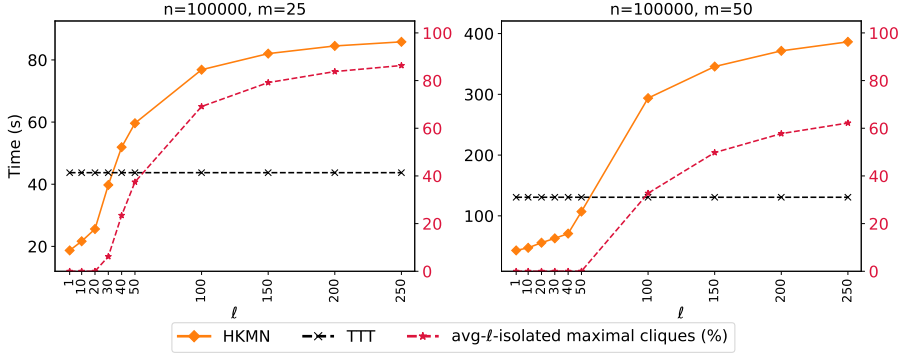


Figure 4.12: Comparison between HKMN and TTT on $BA_{n,m}$ synthetic graphs for avg- ℓ -isolation. The dashed red line shows the number of avg- ℓ -isolated maximal cliques relative to the total number.

4. FILTERING THE CLIQUE EXPLOSION: ENUMERATING ℓ -ISOLATED MAXIMAL CLIQUES EFFICIENTLY

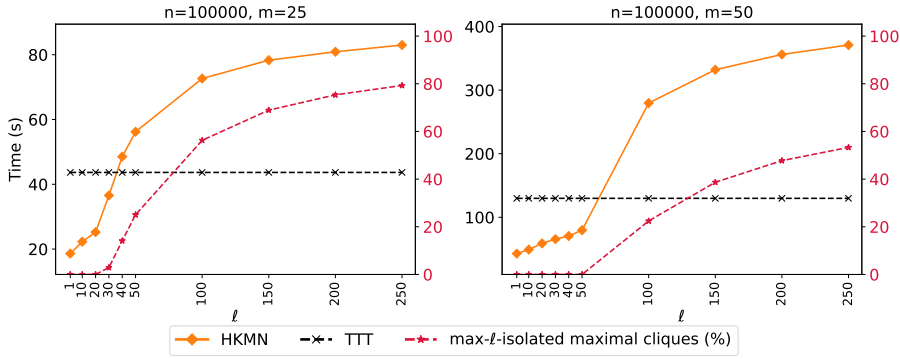


Figure 4.13: Comparison between HKMN and TTT on $BA_{n,m}$ synthetic graphs for max- ℓ -isolation. The dashed red line shows the number of max- ℓ -isolated maximal cliques relative to the total number.

Conclusions and Open Problems

The first part of this thesis investigated how maximal cliques can be leveraged as fundamental building blocks for the analysis of complex networks, addressing both their role in supporting community detection ([Question 0.1](#)) and the problem of meaningfully summarizing them ([Question 0.2](#)). A common theme across [Chapters 2](#) to [4](#) is the contrast between theoretical hardness and practical effectiveness. Although maximal cliques may be exponentially many and difficult to handle, the results presented here show that, on sparse real-world graphs, clique-based approaches can be deployed effectively when supported by suitable algorithmic techniques. Overall, these chapters demonstrate that maximal cliques constitute a powerful and versatile abstraction for modeling induced subgraphs in graph analysis, provided that their redundancy and abundance are carefully managed.

We now summarize the main contributions and open problems of each chapter.

Community detection via maximal cliques. In [Chapter 2](#), we addressed [Question 0.1](#) by introducing CLIQUE-TF-IDF, a novel framework for graph partitioning that describes each vertex through its participation in maximal cliques. The approach builds a positional embedding based on clique membership and is flexible with respect to the clustering algorithm applied to the embedding, supporting both scenarios in which the number k of communities is known in advance and those in which it is not.

The higher computational cost of CLIQUE-TF-IDF represents a deliberate trade-off for partition quality. Experimental results show that, when k is

5. CONCLUSIONS AND OPEN PROBLEMS

fixed, the method consistently outperforms state-of-the-art competitors, while remaining competitive even when k is unknown. Although the framework relies on the enumeration of maximal cliques, existing results and our experiments confirm that clique enumeration is feasible on sparse real-world graphs. In practice, the main limiting factor is the number of cliques to be processed, rather than their computation, as also observed in [Section 2.6.1](#). Within these constraints, our implementation scales to graphs with tens of thousands of vertices and hundreds of thousands of edges on modest hardware.

Beyond community detection, this chapter highlights the potential of hybrid approaches combining combinatorial algorithms with AI-inspired techniques. Similar strategies could be applied to other intractable problems where informative substructures can be efficiently enumerated. For example, listing independent sets can aid vertex coloring, as each independent set corresponds to a maximal clique in the complement graph, and max-cut problems may benefit from computing maximal bipartite subgraphs [\[174\]](#). These hybrid methods exploit the strengths of both combinatorial and AI techniques and may open new multidisciplinary research directions.

We also identify several promising directions for future work:

- Reducing the high running time when the input network contains many maximal cliques, for instance via summarization techniques as discussed in [Chapter 3](#).
- Implementing distributed versions of the approach using distributed maximal clique enumeration algorithms [\[26, 27, 28, 29\]](#).
- Exploring other dense substructures, such as trusses [\[175\]](#), k -plexes [\[36\]](#), fixed-size cliques [\[90, 176\]](#), or k -diamonds [\[177\]](#), as the basis for vertex embeddings to potentially improve partition quality.
- Applying similar strategies to compute structural clusterings [\[55, 178, 54\]](#).

Maximal clique summarization. In [Chapter 3](#), we addressed [Question 0.2](#) by introducing a dual-axis taxonomy for maximal clique summarization, based on summarization principles and problem classes. The literature on this topic is highly scattered: contributions often lack cross-references, show limited awareness of the broader landscape, and miss opportunities to exploit valuable techniques from related works. Our framework unifies existing approaches, reveals connections between seemingly disparate problems, and provides a conceptual basis for further study. To the best of our knowledge, this is the first

5. CONCLUSIONS AND OPEN PROBLEMS

attempt to offer a comprehensive view and comparison of different maximal clique summarization problems.

Several versions of maximal clique summarization (e.g., [Problem 3.6](#) and [Problem 3.11](#)) have a well-established literature, mostly devoted to combinatorial results [[106](#), [107](#), [122](#)]. Here, we focused on the algorithmic aspects of maximal clique summarization, emphasizing methods and approaches from the literature. Most contributions date back to the last decade, and many problems are still open or have not been addressed at all (see [Tables 3.1](#) to [3.3](#)). Investigating such open problems would benefit both the theoretical understanding of maximal clique summarization and its application in practical contexts.

Our work opens several promising research directions:

- Many problems within our framework with practical applications remain unexplored. Studying these open problems would significantly advance the literature on clique summarization.
- Efficiency, especially for Search and Reduced Size problems, has received limited attention. New algorithms could improve performance on large graphs and produce smaller summaries.
- Scalability could be enhanced via alternative computational models, such as cache-efficient, external memory, parallel, or distributed algorithms.
- Streaming versions of these problems should be investigated to handle dynamic graphs, e.g., in social or communication networks.
- Algorithms exploiting special graph properties (e.g., sparseness, scale-freeness, hierarchical structure, small-world, high clustering) may have practical impact.
- There is a lack of tools for experimental analysis, open-source software, and shared benchmarks with known optimal or high-quality summaries.

Enumerating ℓ -isolated maximal cliques. In [Chapter 4](#), we addressed [Question 0.2](#), focusing on a specific setting of maximal clique summarization: the enumeration of ℓ -isolated maximal cliques. We introduced four pruning heuristics that can be applied on top of classical maximal clique enumeration algorithms. We have theoretically proved their correctness and conducted an extensive experimental analysis on a variety of real-world and synthetic graphs, demonstrating the effectiveness of two of our variants against state-of-the-art competitors. We also established a computational hardness result,

5. CONCLUSIONS AND OPEN PROBLEMS

showing that when ℓ is part of the input, finding an ℓ -isolated maximal clique is NP-complete; as a consequence, unless $P = NP$, no output-sensitive polynomial-time enumeration algorithm exists, i.e., no algorithm whose running time is polynomial in the number of ℓ -isolated maximal cliques.

Overall, ℓ -isolation appears to be a useful lens for filtering cliques according to how strongly they are separated from the rest of the network. Small values of ℓ expose only the most tightly isolated structures, at the cost of excluding larger cliques. At the other extreme, too large values of ℓ may contribute little beyond the raw clique distribution, as almost every clique qualifies. The most informative patterns emerge at intermediate thresholds, such as $\ell = 50$ or 100 , which strike a balance between selectivity and inclusiveness: they admit a substantial fraction of cliques, even larger ones, while still preserving the notion of relative isolation. A natural way to address the trade-off between selectivity and clique size, which is closely tied to the choice of ℓ , could be to introduce alternative metrics in which the isolation factor is defined relative to clique size, rather than as an absolute threshold.

Isolation, and clique summarization more broadly, pose many other open problems. In particular, it would be worthwhile to investigate whether our approach can be extended to the computation of ℓ -isolated cliques by size (see [Problem 3.3](#)). Another promising direction is to examine whether the problem admits fixed-parameter tractable algorithms under alternative parameters, such as degeneracy, either by refining the techniques of [\[161, 162\]](#) or by focusing on specific graph families with distinctive structural properties.

Part II

Bounded Degree and Outerplanar Induced Subgraphs in Planar Graphs

Chapter 6

Introduction to Large Constrained Induced Subgraphs in Planar Graphs

The study of induced subgraphs has a central role in Graph Theory. As a prominent example, establishing the truth of the Albertson and Berman conjecture [22, 21], stating that every n -vertex planar graph contains an induced forest with at least $\frac{n}{2}$ vertices, is one of the most famous and frequently mentioned (see, e.g., [179, 15, 180, 181, 182, 183, 184, 23, 185, 186, 187]) open problems in Graph Theory. The conjecture is the strongest possible, as there exist n -vertex planar graphs that have no set with more than $n/2$ vertices inducing a forest (for example, a planar graph composed of $n/4$ vertex-disjoint K_4 's). The conjecture has been proved for many classes of planar graphs, including outerplanar graphs [188], triangle-free planar graphs [180, 184, 185], n -vertex planar graphs with at most $\lfloor 7n/4 \rfloor$ edges [186], bipartite planar graphs [187], and 2-outerplanar graphs [15], often with bounds that are even stronger than $n/2$. A positive solution to Albertson and Berman's conjecture would imply, without using the 4-color theorem, the existence of an independent set with $n/4$ vertices in any n -vertex planar graph.

The best known lower bound for the size of a set of vertices that induces a forest in any n -vertex planar graph is $2n/5$. This is a consequence of a result of Borodin [189], which asserts that every n -vertex planar graph has an acyclic 5-coloring; in such a coloring, the two largest color classes induce a forest with at least $2n/5$ vertices. Since there exist planar graphs with no acyclic 4-coloring, this line of attack to the Albertson and Berman's conjecture

6. INTRODUCTION TO LARGE CONSTRAINED INDUCED SUBGRAPHS IN PLANAR GRAPHS

cannot yield bounds better than $2n/5$. A promising alternative direction is the one of finding, in any n -vertex planar graph G , a large induced subgraph which belongs to a restricted planar graph class, for which it is known that a large induced forest can always be found. In particular, it is known that every n -vertex outerplanar graph has an induced forest with at least $2n/3$ vertices [188], and thus every n -vertex planar graph has an induced forest with at least $2cn/3$ vertices, where c is the maximum constant such that every n -vertex planar graph has an induced outerplanar graph with at least cn vertices. Hence, any bound for c larger than $3/5$ would improve the currently best known bound of $2n/5$ for the Albertson and Berman's conjecture. It is not hard to observe that $c \leq 2/3$ (as one of the two 6-vertex maximal planar graphs has a largest induced outerplanar graph with 4 vertices only) and that $c \geq 1/2$ (this comes from a decomposition in *outerplanar layers* of the given graph). The best known lower bound for c is $11/21$, due to Bose, Dujmović, Houdrouge, Morin, and Odak [190]. In fact, the $11n/21$ lower bound [190], as well as the previous $n/2$ lower bound [191, 179] holds for the size of an *outerplane graph* that one is guaranteed to find in an n -vertex planar graph. That is, fix a plane embedding of the n -vertex planar graph; then the found subset induces an embedded planar graph in which every vertex is incident to the outer face. This property allows the result to be used for graph drawing applications, see [179, 190, 192].

Other popular related conjectures state that every n -vertex bipartite planar graph has an induced forest with at least $\frac{5n}{8}$ vertices [22] and that every n -vertex planar graph has an induced linear forest with at least $\frac{4n}{9}$ vertices [23]. In the former case, the best known lower bound is $\lceil \frac{4n+3}{7} \rceil$ [187], while in the latter case it is $\lceil \frac{n}{3} \rceil$ [193].

Combinatorial problems on induced graphs usually have the following form: Given a class $\mathcal{G}$ of n -vertex graphs, what is the largest function $f(n)$ such that every graph in $\mathcal{G}$ has an induced subgraph with $f(n)$ vertices and satisfying certain properties? Since in a clique one cannot find any induced subgraph other than a clique, the class $\mathcal{G}$ usually contains sparse graphs – most often $\mathcal{G}$ is the class of the n -vertex planar graphs or a subclass of it. More specifically, research on this topic has addressed the problem of determining the maximum size of: (i) an induced forest that one is guaranteed to find in a planar graph [22, 21, 186], in an outerplanar graph [188], in a 2-outerplanar graph [15], in a planar graph with given girth [194, 180, 195, 183, 184, 185], in a graph with bounded treewidth [196], or in a cubic graph [197, 198, 199]; (ii) an induced linear forest that one is guaranteed to find in a planar graph [23, 193], in an outerplanar graph [23], or in a planar graph with given girth [181]; (iii) a k -degenerate induced graph that one is guaranteed to find in a planar graph [200, 201, 182];

6. INTRODUCTION TO LARGE CONSTRAINED INDUCED SUBGRAPHS IN PLANAR GRAPHS

and (iv) an induced matching that one is guaranteed to find in a planar graph with degree at most 3 [202].

Let $\mathcal{F}$ denote the class of constrained induced subgraphs that we aim to find in $\mathcal{G}$. In the remainder of this part, we study lower and upper bounds on the value of $f(n)$ (thus addressing Question 0.3) in the following settings: (i) $\mathcal{F}$ is the class of graphs with maximum degree at most k ($k \geq 3$), and $\mathcal{G}$ is either the class of n -vertex planar graphs or the class of n -vertex outerplanar graphs; in this case, we denote the corresponding function by $f_k(n)$ (Chapter 7); (ii) $\mathcal{F}$ is the class of outerplanar graphs and $\mathcal{G}$ is the class of n -vertex 2-outerplanar graphs (Chapter 8); and (iii) $\mathcal{F}$ is the class of outerplanar graphs and $\mathcal{G}$ is the class of n -vertex planar 3-trees (Chapter 9).

Large Induced Subgraphs of Bounded Degree in Outerplanar and Planar Graphs

7.1 Introduction

In this chapter¹, we study the maximum size of an induced graph of bounded degree that one is guaranteed to find in an outerplanar graph or in a planar graph. It was surprising to us, given that the degree is one of the most important, natural, and used graph parameters, that this problem has not been studied systematically before, to the best of our knowledge. A set of vertices which induces a graph of degree at most k in a graph G is sometimes called a *k -stable set* [204]. Our results are as follows.

- We show that every n -vertex outerplanar graph has an induced subgraph with degree at most 3 and with at least $\frac{2}{3}n$ vertices, an induced subgraph with degree at most k and with at least $\frac{2k+1}{2k+5}n$ vertices, for any even integer $k \geq 4$, and an induced subgraph with degree at most k and with at least $\frac{2k+2/3}{2k+14/3}n$ vertices, for any odd integer $k \geq 5$. On the other hand, for any integer $k \geq 3$, there exists an n -vertex outerplanar graph, in fact

¹The content of this chapter appeared in [203] and is joint work with Fabrizio Frati. We thank Ross J. Kang and Michael J. Pelsmayer for pointing out relevant literature related to the problem studied.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

	Forests	Outerplanar graphs	Planar graphs
LB	0.75 [196]	0.6 [Theorem 7.5]	> 0.384 [Theorem 7.8]
UB	0.75 [196]	0.6 [Theorem 7.1]	< 0.572 [Theorem 7.7]

Table 7.1: Summary of the best known coefficients of the linear term for $k = 3$.

an n -vertex outerpath, such that any induced subgraph with degree at most k has at most $\frac{k-1}{k}n + O(1)$ vertices. The last bound is tight for the class of the n -vertex outerpaths.

- We show that every n -vertex planar graph has an induced subgraph with degree at most 3 and with at least $\frac{5}{13}n$ vertices, an induced subgraph with degree at most 4 and with at least $\frac{27}{59}n$ vertices, and, for any integer $k \geq 5$, an induced subgraph with degree at most k and with at least $\frac{k-2}{k+1}n$ vertices. These results actually follow from bounds with respect to the number of vertices *and edges* for (not necessarily planar) graphs. On the other hand, for any integer $k \geq 3$, there exists an n -vertex planar graph such that any induced subgraph with degree at most k has at most $\frac{2k-2}{2k+1}n + O(1)$ vertices and an n -vertex planar graph such that any induced subgraph with degree at most k has at most $\frac{k+2}{k+4}n + O(1)$ vertices.

Table 7.1 shows the coefficients of the linear terms in the size of the induced subgraphs of degree at most 3 that we can find in the considered graph classes.

Related work. Although, as far as we know, the question of determining the maximum size of an induced subgraph of bounded degree that one is guaranteed to find in an outerplanar graph or in a planar graph was not addressed explicitly before, some results from the literature have an implication or a relationship to our problem. Interestingly, linear lower bounds on the size of such subgraphs can be derived from known results on different graph parameters. However, our lower bounds have larger multiplicative coefficient of the linear term; making this coefficient as large as possible is the typical objective in this research field, think again about the Albritson and Berman conjecture [22, 21] and the related results mentioned above.

- For $k = 0$, a k -stable set is just called *stable set* or *independent set*; the literature about finding large independent sets in planar graphs is one among the richest in the entire field of graph theory, see, e.g. [205, 206, 207, 208, 209, 210]. In particular, a consequence of the four-color theorem is that every planar graph admits an independent set with $\frac{n}{4}$ vertices.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

This defines an induced graph whose degree is bounded by k , for any $k \geq 0$.

- Fountoulakis, Kang, and McDiarmid [204] studied the maximum size of an induced subgraph of bounded degree that one is expected to find in a random graph.
- A (c, k) -defective coloring of a graph is a coloring of the vertices with c colors such that each vertex v has at most k neighbors with the same color as v . Since the vertices in each color class induce a graph whose degree is bounded by k , the existence of a (c, k) -defective coloring for a graph G implies the existence of a set of $\frac{n}{c}$ vertices that induce a subgraph of G with degree at most k . Cowen, Cowen, and Woodall [211] proved that every outerplanar graph has a $(2, 2)$ -defective coloring and that every planar graph has a $(3, 2)$ -defective coloring. These results imply the existence of sets with $\frac{n}{2}$ and $\frac{n}{3}$ of vertices in an n -vertex outerplanar or planar graph, respectively, that induce graphs with degree at most 2. Our lower bounds are larger than these values, already for $k = 3$ in which they are $\frac{2}{3}n$ and $\frac{5}{13}n$, respectively. See [212] for many more results on (c, k) -defective colorings.
- Linear forests have degree at most 2, hence the size of an induced linear forest that one is guaranteed to find in an n -vertex outerplanar graph (the optimal size $\lceil \frac{4n+2}{7} \rceil$ was established by Pelsmayer [23]) or in an n -vertex planar graph (the best known lower bound $\lceil \frac{n}{3} \rceil$ is due to Poh [193]) is a lower bound for the size of an induced graph with degree at most k , for any integer $k \geq 3$, that one is guaranteed to find in an n -vertex outerplanar or planar graph, respectively. Our lower bounds are larger than these values, already for $k = 3$ in which they are $\frac{2}{3}n$ and $\frac{5}{13}n$, respectively.
- Chappell and Pelsmayer [196] proved that every n -vertex graph with treewidth t contains an induced forest with degree at most k and with at least $\lceil \frac{2kn+2}{tk+k+1} \rceil$ vertices. This provides a lower bound of $\lceil \frac{2kn+2}{3k+1} \rceil$ on the size of an induced subgraph with degree at most k that one is guaranteed to find in an n -vertex outerplanar graph. Our lower bounds are larger than $\lceil \frac{2kn+2}{3k+1} \rceil$ for any $k \geq 3$ (and sufficiently large n).
- Chappell and Pelsmayer [196] also proved (and attributed the result to Chappell, Gimbel, and Hartman) that every n -vertex forest contains an induced forest with degree at most k and with at least $\lceil \frac{k+1}{k+2}n \rceil$ vertices. This bound is the best possible.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

- It can be proved by edge-counting arguments that sparse graphs, like planar graphs, have many vertices (and hence large induced subgraphs) with bounded degree. The linear lower bounds one gets in this way are worse than those of this chapter, see, e.g., [213, 214]. In the same spirit, Bose, Dujmović, and Wood [215] considered the following problem. Let $\mathcal{G}_{n,w}$ be the family of the n -vertex graphs with treewidth w , where $n \geq 2w + 1$. For given integers $0 \leq t \leq w$ and $k \geq 2w$, what is the largest size of an induced graph that one is guaranteed to find in every graph G in $\mathcal{G}_{n,w}$ such that the treewidth of the induced graph is at most t and the degree of its vertices in G is at most k ? They provided a bound for this problem which implies that every n -vertex outerplanar graph G (in fact, every graph with treewidth 2) has a subset of at least $\frac{k-3}{k-1}n$ vertices whose degree in G is at most k . This lower bound is weaker than our lower bounds for every integer $k \geq 3$.
- Knauer and Ueckerdt [216] considered the following problem. For given integers $w \geq 1$ and $c \geq 1$, what is the largest size of an induced graph that one is guaranteed to find in every graph of treewidth at most w such that each connected component of the induced graph has at most c vertices? They provided a $\frac{c}{w+c+1}n$ lower bound and a $\frac{c}{w+c}n$ upper bound for this problem. Their results imply (by setting $c = k + 1$) a $\frac{k+1}{w+k+2}n$ lower bound for the size of an induced graph of degree at most k that one is guaranteed to find in every graph of treewidth at most w . In particular, for outerplanar graphs this yields a $\frac{k+1}{k+4}n$ lower bound, which is weaker than our lower bounds for every integer $k \geq 3$.
- Finally, as mentioned above, the size of k -degenerate subgraphs in planar graphs, for $k = 2$ [200], $k = 3$ [201], and $k = 4$ [182] has been studied. A lower bound on the size of an induced subgraph with degree at most k is also a lower bound on the size of a k -degenerate induced subgraph, and an upper bound for the latter size is an upper bound for the former one. The upper and lower bounds in [201, 182] are in fact larger than ours.

Chapter organization. The rest of the chapter is organized as follows. In Section 7.2 we give some preliminaries; in Section 7.3 and Section 7.4, we present our results for outerplanar graphs and planar graphs, respectively.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

7.2 Preliminaries

We refer to [Chapter 1](#) for basic definitions on graphs. Let $G = (V, E)$ be a simple undirected graph. We recall that, for any set $\mathcal{I} \subseteq V(G)$, $G[\mathcal{I}]$ denotes the subgraph of G induced by $\mathcal{I}$, and that the degree of G is the maximum degree among its vertices. A set $\mathcal{D} \subseteq V(G)$ is a *dominating set* if every vertex of G is in $\mathcal{D}$ or is adjacent to a vertex in $\mathcal{D}$. We also recall that a planar drawing is outerplanar if every vertex is incident to the outer face; an outerplanar graph is a graph that admits an outerplanar drawing; the topological information associated with such a drawing is called an outerplane embedding; and a graph together with an outerplane embedding is called an outerplane graph. In the following, we denote by $\mathcal{O}_G$ the outerplane embedding of a maximal outerplanar graph G . Without loss of information, one can think that $\mathcal{O}_G$ is just an outerplanar drawing of G . Observe that the outer face of $\mathcal{O}_G$ is delimited by a Hamiltonian cycle, while the internal faces are delimited by cycles with 3 vertices. An edge uv of G splits G into two outerplanar graphs, namely the subgraphs of G that are induced by the vertices encountered when walking in clockwise and counter-clockwise direction along the outer face of $\mathcal{O}_G$ from u to v . These are called *uv-split subgraphs* of G . The vertices u and v , as well as the edge uv , belong to both *uv-split subgraphs* of G . If uv is incident to the outer face of $\mathcal{O}_G$, then one of these subgraphs is just the edge uv and the other one is G . We define the *distance* in $\mathcal{O}_G$ between two distinct edges uv and xy as the number of steps that are needed to reach xy from uv , where a step moves an edge wz (initially $wz := uv$) to an edge that appears together with wz on the boundary of an internal face of $\mathcal{O}_G$. Thus, for example, if uv and xy are incident to the same internal face of $\mathcal{O}_G$, their distance in $\mathcal{O}_G$ is 1. The *xy-parent* of the edge uv is the edge rs that appears together with uv on the boundary of an internal face of $\mathcal{O}_G$ and that is such that the distance between xy and uv in $\mathcal{O}_G$ is 1 plus the distance between xy and rs in $\mathcal{O}_G$. Finally, observe that the weak dual of $\mathcal{O}_G$ is a tree with degree at most 3. We say that G is a *maximal outerpath* if its weak dual is a path. An *outerpath* is a subgraph of a maximal outerpath.

7.3 Induced Subgraphs of Outerplanar Graphs

In this section, we consider the size of the largest induced subgraph of degree at most k that one is guaranteed to find in an n -vertex outerplanar graph. [Table 7.2](#) shows the coefficients of the linear terms in the size of the induced

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

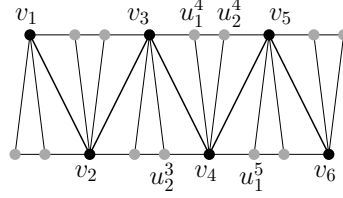


Figure 7.1: Illustration for [Theorem 7.1](#), with $n = 18$, $k = 3$, and $h = 6$.

subgraphs we can find, for various values of k .

	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
LB	$\frac{6}{7}$	> 0.69	0.72	> 0.76	> 0.78
UB	$\frac{6}{7}$	0.75	0.8	0.83	< 0.858

Table 7.2: Results on outerplanar graphs with different values of k .

We start with an upper bound, which comes from a family of outerpaths, see [Figure 7.1](#).

Theorem 7.1. *For every $n \geq 1$ and for every integer $k \geq 3$, there exists an n -vertex outerpath G with the following property. Consider any set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k . Then $|\mathcal{I}| \leq \frac{k-1}{k}n + 2$.*

Proof. Let G be the n -vertex outerpath depicted in [Figure 7.1](#) and defined as follows. Let $h = \lfloor \frac{n}{k} \rfloor$. The graph contains a path $v_1, \dots, v_h$, vertices $u_1^i, \dots, u_{k-1}^i$, for $i = 1, \dots, h$, and edges $u_j^i u_{j+1}^i$, $v_i u_j^i$, $v_{i-1} u_1^i$, and $u_{k-1}^i v_{i+1}$, whenever the end-vertices of such edges are in G . Additional $n - h \cdot k$ isolated vertices, which do not play any role in the proof, are added to G , so that it has n vertices.

Consider any set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k . We prove that there exists a set $\mathcal{I}' \subseteq V(G)$ such that $G[\mathcal{I}']$ has degree at most k , such that $|\mathcal{I}'| = |\mathcal{I}|$, and such that $\mathcal{I}'$ does not contain any of $v_2, \dots, v_{h-1}$. Suppose that $\mathcal{I}$ contains a vertex v_i , for some $i \in \{2, \dots, h-1\}$. Then there exists a vertex w among $u_{k-1}^{i-1}, u_1^i, u_2^i, \dots, u_{k-1}^i, u_1^{i+1}$ that is not in $\mathcal{I}$, as otherwise v_i would have degree $k+1$ in $G[\mathcal{I}]$. Replace v_i in $\mathcal{I}$ with w . Clearly, the size of $\mathcal{I}$ remains the same. Also, $G[\mathcal{I}]$ still has degree at most k . Indeed, the only vertices that have degree larger than k in G are $v_1, \dots, v_h$. Also, w is a neighbor of v_i , which however is not in $\mathcal{I}$ after the replacement, and might be a neighbor

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

of v_{i-1} or v_{i+1} . However, v_{i-1} or v_{i+1} are also neighbors of v_i , hence if they are in $\mathcal{I}$, their degree in $G[\mathcal{I}]$ does not increase. The repetition of this replacement results in the desired set $\mathcal{I}'$.

It remains to observe that $|\mathcal{I}'| \leq n - (h - 2) = n - \lfloor \frac{n}{k} \rfloor + 2 \leq \frac{k-1}{k}n + 2$. $\square$

The upper bound of [Theorem 7.1](#) was proved by considering n -vertex out-earpaths. For this graph class, the bound is tight, up to additive constants, as proved in the following theorem.

Theorem 7.2. *For every $n \geq 1$, for every n -vertex out-earpath G , and every integer $k \geq 3$, there exists a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k and $|\mathcal{I}| \geq \frac{k-1}{k}n$.*

Proof. Without loss of generality, we can assume that G is a maximal out-earpath, as otherwise edges can be added to it so that it becomes maximal, and then a subset $\mathcal{I}$ with the properties required by [Theorem 7.2](#) in the augmented graph satisfies the same properties in the original graph. Let e be an edge that is incident both to the outer face of the outerplane embedding $\mathcal{O}_G$ of G and to an internal face corresponding to an extreme of the weak dual of $\mathcal{O}_G$; we refer to e as to the *special edge*. Also, let u be any of the two end-vertices of e , and let v be the other one; we refer to u as to the *special vertex*. We prove, by induction on n , that there exists a set $\mathcal{I} \subseteq V(G)$ such that: (i) $|\mathcal{I}| \geq \frac{k-1}{k}n$; (ii) $G[\mathcal{I}]$ has degree at most k ; and (iii) if the special vertex u is in $\mathcal{I}$, then u has at most $k - 1$ neighbors in $\mathcal{I}$. This implies the statement of the theorem.

There are two base cases:

- **Case 1:** $n \leq k$. In this case, we let $\mathcal{I} := V(G)$. Then $|\mathcal{I}| = n > \frac{k-1}{k}n$. Since there are at most k vertices in $\mathcal{I}$, we have that $G[\mathcal{I}]$ has degree at most $k - 1$.
- **Case 2:** $n = k + 1$. Let x be any vertex of G . We let $\mathcal{I} := V(G) \setminus \{x\}$. Then $|\mathcal{I}| = k > \frac{(k-1)(k+1)}{k} = \frac{k-1}{k}n$. Since there are at most k vertices in $\mathcal{I}$, we have that $G[\mathcal{I}]$ has degree at most $k - 1$.

In the inductive case, we have $n \geq k + 2$. Our strategy is as follows. We identify an edge wz of G with the following properties. Let G_R be the wz -split subgraph of G not containing uv . Also, let G_L be the subgraph of G induced by $V(G) \setminus V(G_R)$. We require that G_L has at least k vertices and that it contains a vertex h such that, by defining $\mathcal{I} = \mathcal{I}_L \cup \mathcal{I}_R$, where $\mathcal{I}_L := V(G_L) \setminus \{h\}$ and $\mathcal{I}_R \subseteq V(G_R)$ is the set obtained by applying induction on G_R , the conditions of the claim are satisfied. We now make this argument precise.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

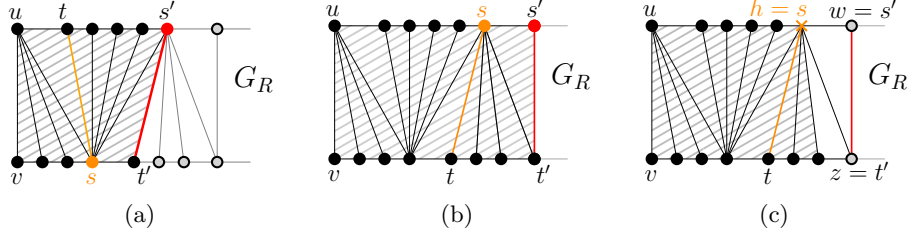


Figure 7.2: Illustration for the proof of [Theorem 7.2](#). Graph G_L is striped, vertices and edges in G_R are either not shown or colored gray, the old active vertex s and edge st are orange, while the new active vertex s' and edge $s't'$ are red. (a) All the neighbors of s are moved into G_L , which still has at most $k + 1$ vertices. (b) All the neighbors of s are moved into G_L , which now has at least $k + 2$ vertices. (c) The desired vertex h is set to be s , the desired edge wz is set to be $s't'$, s' and t' are moved to G_R , $\mathcal{I}_L = V(G_L) \setminus \{h\}$, and induction is applied on G_R with wz as a special edge.

In order to find the desired edge wz , subgraph G_L , and vertex h , we work iteratively. We initialize G_L to consist of the vertices u and v (and of the edge uv), and G_R to $G[V(G) \setminus \{u, v\}]$. Notice that only one of u and v has at least two neighbors in G_R ; such a vertex is called *active vertex* and uv is the *active edge*. While G_L has at most $k + 1$ vertices, let s be the current active vertex and let st be the current active edge. We move to G_L all the neighbors of s that are in G_R . Let s' and t' be the neighbors of s such that the edge $s't'$ exists and its distance from uv in $\mathcal{O}_G$ is maximum. Now two cases are possible. First, if G_L still has at most $k + 1$ vertices, as in [Figure 7.2\(a\)](#), we continue the iteration. Only one of s' and t' has at least two neighbors in G_R ; such a vertex is the new active vertex and $s't'$ is the new active edge. Conversely, if G_L has at least $k + 2$ vertices, as in [Figure 7.2\(b\)](#), we stop the iteration, we move s' and t' back to G_R , as in [Figure 7.2\(c\)](#), we define the desired vertex h to be s , the desired edge wz to be $s't'$, and the desired graph G_L to be the current graph G_L . We define $\mathcal{I}_L = V(G_L) \setminus \{h\}$. We apply induction on G_R with wz as a special edge and with the one between w and z that has more than two neighbors in G_R as a special vertex. The induction returns us a set $\mathcal{I}_R \subseteq V(G_R)$ and we define $\mathcal{I} := \mathcal{I}_L \cup \mathcal{I}_R$.

We now show that the set $\mathcal{I}$ satisfies the required properties.

- First, we prove that $|\mathcal{I}| \geq \frac{k-1}{k}n$. By induction, we have $|\mathcal{I}_R| \geq \frac{k-1}{k}|V(G_R)|$. Also, $|V(G_L)|$ is at least k , given that it is at least $k + 2$ before s' and

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

t' are moved back to G_R . Since $|\mathcal{I}_L| = |V(G_L)| - 1$, we have that $|\mathcal{I}_L| \geq \frac{k-1}{k}|V(G_L)|$. Hence $|\mathcal{I}| = |\mathcal{I}_L| + |\mathcal{I}_R| \geq \frac{k-1}{k}(|V(G_L)| + |V(G_R)|) = \frac{k-1}{k}n$.

- Second, we prove that, if $u \in \mathcal{I}$, then u has at most $k - 1$ neighbors in $\mathcal{I}$. Notice that u can have at least k neighbors in $\mathcal{I}$ only if it has degree at least $k \geq 3$. This implies that u is selected as the active vertex in the first step of the iteration for the definition of G_L . If u has degree at least $k + 1$, then the iteration defining G_L stops at the first step by inserting in G_L all the neighbors of u , which implies that $h = u$ and that u is not in $\mathcal{I}$. However, if u has degree k , then after inserting in G_L all the neighbors of u , we have that G_L has $k + 1$ vertices. Now the active vertex becomes one of u 's neighbors, which is then selected as h , and hence not inserted in $\mathcal{I}$, given that the second step of the iteration results in G_L having at least $k + 2$ vertices. Hence, u has $k - 1$ neighbors in $\mathcal{I}$.
- Finally, we prove that $G[\mathcal{I}]$ has degree at most k . Since w and z are the only vertices of G_R with neighbors in G_L , we have that any vertex in $V(G_R) \setminus \{w, z\}$ has degree at most k in $G[\mathcal{I}]$, by induction. Next, consider the graph G_L and the active edge st in the step of the iteration defining G_L before the last one, i.e., before all the neighbors of s in G_R are moved to G_L and two of them are moved back to G_R since G_L has at least $k + 2$ vertices. At that point, G_L has at most $k + 1$ vertices, hence all its vertices other than s and t have degree at most k in G and hence also in $G[\mathcal{I}]$. Furthermore, s is not in $\mathcal{I}$, since at the next step of the iteration it is selected as the vertex h . Also, t has at most one neighbor not in G_L , since it is not the active vertex, which bounds its degree in $G[\mathcal{I}]$ to k , given that out of its potential k neighbors in G_L , one of them, namely s , is not inserted into $\mathcal{I}$. All the vertices that are inserted into G_L at the last step of the iteration, other than w and z , have degree 3 in G , and hence at most 3 in $G[\mathcal{I}]$. The vertex between w and z that is not selected as the special vertex has two neighbors in G_R and at most two neighbors in G_L , however one of the two neighbors in G_L is h , hence that vertex has degree at most $3 \leq k$ in $G[\mathcal{I}]$. Finally, the vertex between w and z that is selected as the special vertex has at most $k - 1$ neighbors from G_R in $\mathcal{I}$ (this is where we use the condition on the degree of the special vertex), and at most two neighbors in G_L , however one of the two neighbors in G_L is h , hence that vertex has degree at most k in $G[\mathcal{I}]$.

This concludes the proof of the theorem. $\square$

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

We conclude the section by showing lower bounds for the size of an induced subgraph of degree at most k that one is guaranteed to find in an outerplanar graph. We exhibit three lower bounds, one for $k = 3$, one for $k \geq 4$ with k even, and one for $k \geq 5$ with k odd. The first lower bound is tight, because of [Theorem 7.1](#). Before stating the theorems, we present some tools that we are going to use for their proofs. By the same reasoning used for outerpaths, we can assume that the input is a maximal outerplanar graph G . Our first tool is a decomposition lemma, which is stated in the following.

Lemma 7.3. *Let G be an n -vertex maximal outerplanar graph, let $\ell \geq 1$ be an integer, and let xy be an edge incident to the outer face of $\mathcal{O}_G$. If $n \geq \ell + 2$, there exists an edge $uv \in E(G)$ such that a uv -split subgraph H of G satisfies the following properties:*

- H has a number h of vertices such that $\ell + 2 \leq h \leq 2\ell + 1$; and
- if uv does not coincide with the edge xy , then H does not contain the edge xy .

Proof. The proof is by induction on n . In the base case we have $\ell + 2 \leq n \leq 2\ell + 1$. Then let uv coincide with xy , thus satisfying the second property. Also, uv splits G into two outerplanar graphs, one of which is G and has hence the required number of vertices.

If $n \geq 2\ell + 2$, let z be the vertex such that the cycle (x, y, z) bounds an internal face of $\mathcal{O}_G$. Let G_1 be the xz -split subgraph of G that does not contain y . Let G_2 be the yz -split subgraph of G that does not contain x . We distinguish two cases.

- If both G_1 and G_2 have less than $2\ell + 2$ vertices, then at least one of them has at least $\ell + 2$ vertices, given that $|V(G_1)| + |V(G_2)| = n + 1 \geq 2\ell + 3$, where the first equality is due to the fact that G_1 and G_2 share z . Then xz or yz is the required edge, depending on whether G_1 or G_2 has at least $\ell + 2$ vertices, respectively.
- Otherwise, one of G_1 and G_2 , say G_1 , has at least $2\ell + 2$ vertices. See [Figure 7.3](#). By induction, G_1 contains an edge uv such that a uv -split subgraph H of G_1 has a number h of vertices such that $\ell + 2 \leq h \leq 2\ell + 1$ and does not contain the edge xz . The latter condition implies that H is also one of the two uv -split subgraphs of G and that H does not contain the edge xy , as required.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

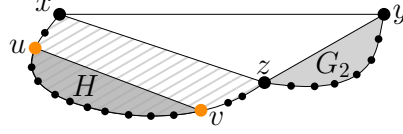


Figure 7.3: Illustration for the proof of [Lemma 7.3](#). The case in which G_1 has at least $2\ell + 2$ vertices. Most internal edges of G are not shown, the interior of G_1 is striped.

This concludes the proof of the lemma. $\square$

For a subgraph H of G , we denote by $\overline{H}$ the subgraph of G induced by $V(G) \setminus V(H)$. Note that $\overline{H}$ is a (not necessarily maximal) outerplanar graph. The basic strategy we use in order to find a large set $\mathcal{I} \subseteq V(G)$ of vertices such that $G[\mathcal{I}]$ has degree at most k is the following. By employing [Lemma 7.3](#), we find an edge uv such that, in a uv -split subgraph H of G , we can select a set $\mathcal{I}_H \subset V(H)$ of vertices that contains the fraction of the vertices in $V(H)$ we are aiming for ($\frac{2}{3}$ for $k = 3$, $\frac{2k+1}{2k+5}$ for $k \geq 4$ with k even, and $\frac{2k+2/3}{2k+14/3}$ for $k \geq 5$ with k odd), that induces a subgraph of G with degree at most k , and that does not contain u and v . We also apply induction to find a large set of vertices $\mathcal{I}_{\overline{H}}$ in $\overline{H}$ such that $G[\mathcal{I}_{\overline{H}}]$ has degree at most k . Then the set $\mathcal{I} = \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ has the required number of vertices and $G[\mathcal{I}]$ has degree at most k . This is formalized in the following lemma.

Lemma 7.4. *Let $k \geq 3$ be an integer and let $0 < c < 1$ be a real number. Suppose that G contains an edge uv such that one of the two uv -split subgraphs of G , say H , has a set $\mathcal{I}_H$ of vertices such that $\mathcal{I}_H$ contains neither u nor v , such that $|\mathcal{I}_H| \geq c \cdot |V(H)|$, and such that $G[\mathcal{I}_H]$ has degree at most k . Suppose also that $\overline{H}$ has a set $\mathcal{I}_{\overline{H}}$ of vertices such that $|\mathcal{I}_{\overline{H}}| \geq c \cdot |V(\overline{H})|$ and such that $G[\mathcal{I}_{\overline{H}}]$ has degree at most k .*

Then the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $c \cdot n$ vertices and $G[\mathcal{I}]$ has degree at most k .

Proof. That $\mathcal{I}$ contains at least $c \cdot n$ vertices follows by the assumption on the cardinalities of $\mathcal{I}_H$ and $\mathcal{I}_{\overline{H}}$, and by $|V(H)| + |V(\overline{H})| = n$. That $G[\mathcal{I}]$ has degree at most k follows by the assumption on the degrees of $G[\mathcal{I}_H]$ and $G[\mathcal{I}_{\overline{H}}]$, and by the fact that u and v , which do not belong to $\mathcal{I}_H$, are the only vertices of H with neighbors in $\overline{H}$. $\square$

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

We are now ready to present our theorems. We start with the one for induced subgraphs of degree at most 3.

Theorem 7.5. *For every $n \geq 1$, for every n -vertex outerplanar graph G , there exists a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most 3 and $|\mathcal{I}| \geq \frac{2}{3}n$.*

Proof. Let G be an n -vertex maximal outerplanar graph. The proof is by induction on n . In the base case we have $n \leq 11$. An easy edge-counting argument proves that we can find a set $\mathcal{I}$ with n vertices (if $1 \leq n \leq 4$), with $n - 1$ vertices (if $5 \leq n \leq 6$), with $n - 2$ vertices (if $7 \leq n \leq 8$), or with $n - 3$ vertices (if $9 \leq n \leq 10$) such that $G[\mathcal{I}]$ has degree at most 3. In all four cases the ratio $\frac{|\mathcal{I}|}{n}$ is larger than or equal to $\frac{2}{3}$. The edge-counting argument is as follows. Observe that $\mathcal{O}_G$ has $n - 3$ internal edges and that, for any set $\mathcal{I} \subseteq V(G)$, a vertex with degree at least 4 in $G[\mathcal{I}]$ has at least two incident edges that are internal in $\mathcal{O}_G$. Hence, we can initialize $\mathcal{I} = V(G)$ and then repeatedly remove from $\mathcal{I}$ a vertex with degree at least 4 in $G[\mathcal{I}]$, while such a vertex exists. If $n \leq 4$, then G has at most 1 edge that is internal in $\mathcal{O}_G$, hence no vertex is removed from $\mathcal{I}$. If $5 \leq n \leq 6$, then G has up to 3 edges that are internal in $\mathcal{O}_G$, hence up to 1 vertex is removed from $\mathcal{I}$. If $7 \leq n \leq 8$, then G has up to 5 edges that are internal in $\mathcal{O}_G$, hence up to 2 vertices are removed from $\mathcal{I}$. Finally, if $9 \leq n \leq 10$, then G has up to 7 edges that are internal in $\mathcal{O}_G$, hence up to 3 vertices are removed from $\mathcal{I}$.

If $n = 11$, the above argument requires some amendment, since G has 8 internal edges, hence charging pairs of internal edges to removed degree-4 vertices only upper bounds the number of removed vertices by 4, while we need to show that the number of removed vertices is at most 3 in order to get our desired $\frac{2}{3}$ ratio. The key observation is that, after initializing $\mathcal{I} = V(G)$ and removing from $\mathcal{I}$ the first vertex v with degree at least 4 in G , the number of internal edges in $G[\mathcal{I}]$ decreases at least by three. This is obvious if the degree of v in G is at least 5. If the degree of v in G is 4, let e_1 and e_2 be the internal edges of G incident to v , and let v_0, v_1, v_2 , and v_3 be the neighbors of v , where $e_1 = vv_1$ and $e_2 = vv_2$ lie inside the cycle (v, v_0, v_1, v_2, v_3) of G . Then one of the edges of the path (v_0, v_1, v_2, v_3) is an internal edge e of G , as otherwise G would not be a maximal outerplanar graph or it would contain 5 vertices. Hence, the removal of v from $\mathcal{I}$ not only removes e_1 and e_2 from $G[\mathcal{I}]$, but also makes e an external edge of $G[\mathcal{I}]$. Since the number of internal edges of $G[\mathcal{I}]$ after the removal of v is at most 5, at most 2 more degree-4 vertices are then removed from $G[\mathcal{I}]$.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

Suppose now that $n \geq 12$. As argued before the theorem, it suffices to discuss the case in which G is a maximal outerplanar graph. We define a uv -split subgraph H of G with a suitable size, then we select a large set $\mathcal{I}_H \subseteq V(H)$ of vertices, with $u, v \notin \mathcal{I}_H$, such that $G[\mathcal{I}_H]$ has degree at most 3. Applying induction on $\overline{H}$ to get a set $\mathcal{I}_{\overline{H}}$ and then applying [Lemma 7.4](#) allows us to define the desired set $\mathcal{I}$. We distinguish 16 cases, labeled according to the size of a uv -split subgraph that we can find in G – we need to consider every size in this range. Eventually, we will use [Lemma 7.3](#) with $\ell = 10$, in order to find a uv -split subgraph with a number of vertices between 12 and 21, however for our arguments we need to consider uv -split subgraphs of smaller sizes, as well. Specifically, if we can find a uv -split subgraph H with a number of vertices that is 6, 9, 10, or between 12 and 21, we can select a large set $\mathcal{I}_H \subseteq V(H)$ of vertices, with $u, v \notin \mathcal{I}_H$, such that $G[\mathcal{I}_H]$ has degree at most 3, which allows us to apply [Lemma 7.4](#) to find the desired set $\mathcal{I}$ of vertices in G . If H has 7, 8, or 11 vertices, we can still select a large set $\mathcal{I}_H \subseteq V(H)$ of vertices such that $G[\mathcal{I}_H]$ has degree at most 3, however this requires u and/or v to be part of $\mathcal{I}_H$. The fact that u and/or v belong to $\mathcal{I}_H$ prevents us from employing [Lemma 7.4](#) to find the desired set of vertices in G , however the existence of such a set $\mathcal{I}_H$ is used when uv -split subgraphs with more vertices are considered. In the following, whenever we consider a uv -split subgraph H of G , we denote by w the vertex such that the cycle (u, v, w) bounds an internal face of $\mathcal{O}_H$ and by h the number of vertices of H . Also, we denote by H_1 the uw -split subgraph of G not containing v and by H_2 the vw -split subgraph of G not containing u . We let h_1 and h_2 be the number of vertices of H_1 and H_2 , respectively, where $h_1 \geq 2$, $h_2 \geq 2$, and $h_1 + h_2 = h + 1$, given that w belongs both to H_1 and to H_2 . Finally, we denote by $\mathcal{I}_{\overline{H}} \subseteq V(\overline{H})$ a set such that $G[\mathcal{I}_{\overline{H}}]$ has degree at most 3 and $|\mathcal{I}_{\overline{H}}| \geq \frac{2}{3}|V(\overline{H})|$. This set exists by induction.

Case h=6: Suppose that G contains an edge uv such that a uv -split subgraph H of G has 6 vertices. We define $\mathcal{I}_H := V(H) \setminus \{u, v\}$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{4}{6} = \frac{2}{3}$. Also, $\mathcal{I}_H$ contains 4 vertices, hence $G[\mathcal{I}_H]$ has degree at most 3. By [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired. We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 6 vertices.

Case h=7: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 7$ vertices. Then we have $h_1 \geq 3$, as otherwise H_2 would have 6 vertices. Symmetrically, $h_2 \geq 3$. If there exists a vertex in $V(H) \setminus \{w\}$ that is not adjacent to w , then we define $\mathcal{I}_H := V(H) \setminus \{u, v\}$. Then $|\mathcal{I}_H|/|V(H)| = \frac{5}{7} > \frac{2}{3}$. Also, $G[\mathcal{I}_H]$ has degree at most 3, given that w is adjacent to at most 3 vertices in $\mathcal{I}_H$, by assumption, given that each vertex

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

in $V(H_1) \setminus \{u, w\}$ is not adjacent to at least 1 vertex of H_2 , and given that each vertex in $V(H_2) \setminus \{v, w\}$ is not adjacent to at least 1 vertex of H_1 . By [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

We henceforth assume that, if G has an edge uv such that a uv -split subgraph H of G has 7 vertices, then the degree of w in H is 6. This implies that H admits a *5-over-7 set with u* , a *5-over-7 set with v* , and a *6-over-7 set with u and v* , which are defined as follows. A 5-over-7 set with u is a set $\mathcal{I}_H$ of vertices such that $|\mathcal{I}_H| = 5$, such that $u \in \mathcal{I}_H$ and $v \notin \mathcal{I}_H$, such that $G[\mathcal{I}_H]$ has degree at most 3, and such that the degree of u in $G[\mathcal{I}_H]$ is at most 1. Indeed, the set $\mathcal{I}_H := V(H) \setminus \{v, w\}$ satisfies these properties, see [Figure 7.4\(a\)](#). The definition of a 5-over-7 set with v is analogous, with v in place of u and vice versa, and is realized by the set $\mathcal{I}_H := V(H) \setminus \{u, w\}$, see [Figure 7.4\(b\)](#). A 6-over-7 set with u and v is a set $\mathcal{I}_H$ of vertices such that $|\mathcal{I}_H| = 6$, such that $u, v \in \mathcal{I}_H$, such that $G[\mathcal{I}_H]$ has degree at most 3, and such that the degree of u and v in $G[\mathcal{I}_H]$ is at most 2. Indeed, the set $\mathcal{I}_H := V(H) \setminus \{w\}$ satisfies these properties, see [Figure 7.4\(c\)](#). Although we cannot use such sets to exclude the existence of a uv -split subgraph with 7 vertices, since they contain u or v , we will use them in later cases.

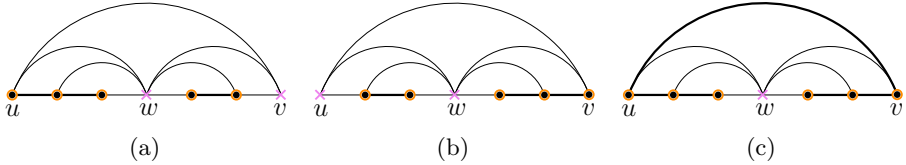


Figure 7.4: A uv -split subgraph H with 7 vertices in which the vertex w has degree 6. (a) A 5-over-7 set with u . (b) A 5-over-7 set with v . (c) A 6-over-7 set with u and v . Vertices in the sets have a circular contour, while vertices not in the sets have a cross.

Case $h=8$: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 8$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge uv such that a uv -split subgraph of G has 6 vertices, we have that either one of h_1 and h_2 is 2 and the other one is 7, or one of h_1 and h_2 is 4 and the other one is 5.

- Suppose that $h_1 = 2$ and $h_2 = 7$, the case in which $h_1 = 7$ and $h_2 = 2$ is symmetric. By assumption, H_2 admits a 5-over-7 set with w , say $\mathcal{I}_{H_2}^w$, and a 6-over-7 set with w and v , say $\mathcal{I}_{H_2}^{wv}$. This implies that H admits a

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

6-over-8 set with u and a *6-over-8 set with v* . The former is a set $\mathcal{I}_H$ of vertices such that $|\mathcal{I}_H| = 6$, such that $u \in \mathcal{I}_H$ and $v \notin \mathcal{I}_H$, such that $G[\mathcal{I}_H]$ has degree at most 3, and such that the degree of u in $G[\mathcal{I}_H]$ is at most 2. Indeed, the set $\mathcal{I}_{H_2}^w \cup \{u\}$ satisfies these properties, see Figure 7.5(a). The definition of a 6-over-8 set with v is analogous, with v in place of u and vice versa, and is realized by the set $\mathcal{I}_{H_2}^{wv}$, see Figure 7.5(b). We also have that H admits a *6-over-8 set with u -deg-1* or a *6-over-8 set with v -deg-1*. The former is a 6-over-8 set with u such that the degree of u in $G[\mathcal{I}_H]$ is at most 1. The latter is a 6-over-8 set with v such that the degree of v in $G[\mathcal{I}_H]$ is at most 1. Indeed, with the current assumption $h_1 = 2$ and $h_2 = 7$, we have that H admits a 6-over-8 set with u -deg-1, while if $h_1 = 7$ and $h_2 = 2$, then H admits a 6-over-8 set with v -deg-1. Although we cannot use such sets to exclude the existence of a uv -split subgraph with 8 vertices, given that they contain u or v , we will use them in later cases.

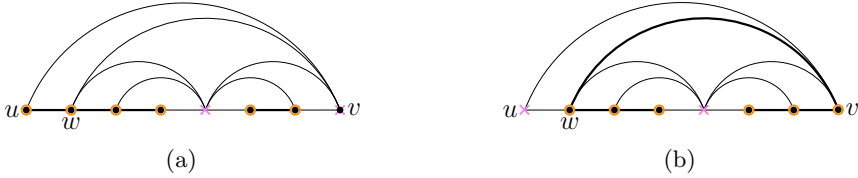


Figure 7.5: A uv -split subgraph H with 8 vertices with $h_1 = 2$ and $h_2 = 7$. (a) A 6-over-8 set with u . (b) A 6-over-8 set with v .

- Suppose that $h_1 = 4$ and $h_2 = 5$, the case in which $h_1 = 5$ and $h_2 = 4$ is symmetric. If w has at most 3 neighbors among the vertices in $V(H) \setminus \{u, v\}$, then we can define $\mathcal{I}_H := V(H) \setminus \{u, v\}$. Then $|\mathcal{I}_H|/|V(H)| = \frac{6}{8} > \frac{2}{3}$. Also, $G[\mathcal{I}_H]$ has degree at most 3. Indeed, consider any vertex $z \in \mathcal{I}_H$. If $z = w$, then z is adjacent to at most 3 vertices in $\mathcal{I}_H$, by assumption. If $z \in V(H_2)$ and $z \neq w$ then z is not adjacent to the 2 vertices of H_1 different from u and w and is not adjacent to z itself. Since u and v are not in $\mathcal{I}_H$, we have that z has at most 3 neighbors in $G[\mathcal{I}_H]$. The argument for the case in which $z \in V(H_1)$ and $z \neq w$ is analogous.

We henceforth assume that, if G has an edge uv such that a uv -split subgraph H of G has 8 vertices and if the uw -split subgraph H_1 has 4 or 5 vertices, then w has degree at least 6 in H . This again implies that H admits a 6-over-8 set with u and a 6-over-8 set with v . Indeed, the former is realized by the set $V(H) \setminus \{v, w\}$, see Figure 7.6(a), and the latter is

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

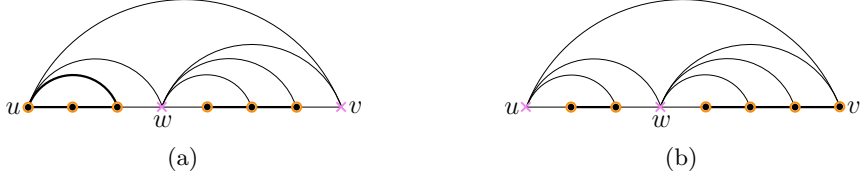


Figure 7.6: A uv -split subgraph H with 8 vertices with $h_1 = 4$ and $h_2 = 5$. (a) A 6-over-8 set with u . (b) A 6-over-8 set with v .

realized by the set $V(H) \setminus \{u, w\}$, see Figure 7.6(b). Also, H admits a 6-over-8 set with u -deg-1 or a 6-over-8 set with v -deg-1. Indeed, with the current assumption $h_1 = 4$ and $h_2 = 5$, we have that H admits a 6-over-8 set with v -deg-1, see again Figure 7.6(b), while if $h_1 = 5$ and $h_2 = 4$ then H admits a 6-over-8 set with u -deg-1.

Case $h=9$: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 9$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 6 vertices, we have that either one of h_1 and h_2 is 2 and the other one is 8, or one of h_1 and h_2 is 3 and the other one is 7, or both h_1 and h_2 are 5.

- Suppose that $h_1 = 2$ and $h_2 = 8$, the case in which $h_1 = 8$ and $h_2 = 2$ is symmetric. By assumption, H_2 admits a 6-over-8 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w$. Since $\mathcal{I}_{H_2}^w$ is a 6-over-8 set with w , we have that $G[\mathcal{I}_H]$ has degree at most 3 and $|\mathcal{I}_H|/|V(H)| = \frac{6}{9} = \frac{2}{3}$. Since u and v do not belong to $\mathcal{I}_H$, by Lemma 7.4, the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.
- Suppose next that $h_1 = 3$ and $h_2 = 7$, the case in which $h_1 = 7$ and $h_2 = 3$ is symmetric. By assumption, H_2 admits a 5-over-7 set with w , say $\mathcal{I}_{H_2}^w$. We define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t\}$, where t is the vertex of H_1 different from u and w . Note that $|\mathcal{I}_H|/|V(H)| = \frac{6}{9} = \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 5-over-7 set with w and since w is the only vertex of H_2 adjacent to t , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 1 in $G[\mathcal{I}_{H_2}^w]$, hence at most 2 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by Lemma 7.4, the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

- Suppose finally that $h_1 = 5$ and $h_2 = 5$. We define $\mathcal{I}_H := V(H) \setminus \{u, v, w\}$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{6}{9} = \frac{2}{3}$. Also, each vertex z in $\mathcal{I}_H \cap V(H_1)$ is adjacent to at most 2 vertices in $\mathcal{I}_H$, namely those in $\mathcal{I}_H \cap V(H_1)$ different from z itself, given that u and w are the only vertices of H_1 adjacent to vertices of H_2 . Analogously, each vertex in $\mathcal{I}_H \cap V(H_2)$ is adjacent to at most 2 vertices in $\mathcal{I}_H$. It follows that $G[\mathcal{I}_H]$ has degree at most 2. By [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 9 vertices.

Case $h=10$: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 10$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 6 or 9 vertices, we have that either one of h_1 and h_2 is 3 and the other one is 8, or one of h_1 and h_2 is 4 and the other one is 7.

- Suppose that $h_1 = 3$ and $h_2 = 8$, the case in which $h_1 = 8$ and $h_2 = 3$ is symmetric. By assumption, H_2 admits a 6-over-8 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t\}$, where t is the vertex of H_1 different from u and w . Note that $|\mathcal{I}_H|/|V(H)| = \frac{7}{10} > \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 6-over-8 set with w and since w is the only vertex of H_2 adjacent to t , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 2 in $G[\mathcal{I}_{H_2}^w]$, hence at most 3 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.
- Suppose that $h_1 = 4$ and $h_2 = 7$, the case in which $h_1 = 7$ and $h_2 = 4$ is symmetric. By assumption, H_2 admits a 5-over-7 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t_1, t_2\}$, where t_1 and t_2 are the vertices of H_1 different from u and w . Note that $|\mathcal{I}_H|/|V(H)| = \frac{7}{10} > \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 5-over-7 set with w and since w is the only vertex of H_2 adjacent to t_1 and t_2 , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 1 in $G[\mathcal{I}_{H_2}^w]$, hence at most 3 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 10 vertices.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

Case $h=11$: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 11$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 6, 9, or 10 vertices, we have that either one of h_1 and h_2 is 4 and the other one is 8, or one of h_1 and h_2 is 5 and the other one is 7.

- Suppose that $h_1 = 4$ and $h_2 = 8$, the case in which $h_1 = 8$ and $h_2 = 4$ is symmetric. By assumption, H_2 admits a 6-over-8 set with w , say $\mathcal{I}_{H_2}^w$, and a 6-over-8 set with v , say $\mathcal{I}_{H_2}^v$. Also by assumption, we have that $\mathcal{I}_{H_2}^w$ is actually a 6-over-8 set with w -deg-1 or that $\mathcal{I}_{H_2}^v$ is actually a 6-over-8 set with v -deg-1.

Suppose first that $\mathcal{I}_{H_2}^w$ is a 6-over-8 set with w -deg-1. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t_1, t_2\}$, where t_1 and t_2 are the vertices of H_1 different from u and w . Note that $|\mathcal{I}_H|/|V(H)| = \frac{8}{11} > \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 6-over-8 set with w -deg-1 and since w is the only vertex of H_2 adjacent to t_1 and t_2 , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 1 in $G[\mathcal{I}_{H_2}^w]$, hence at most 3 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

Suppose next that $\mathcal{I}_{H_2}^v$ is a 6-over-8 set with v -deg-1. If w has at most one neighbor among t_1 and t_2 , then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t_1, t_2\}$ and conclude the induction as above, hence we can assume that w is adjacent to both t_1 and t_2 . This implies that u is a neighbor of at most one of t_1 and t_2 . It follows that H admits a [8-over-11 set with \$u\$](#) and a [8-over-11 set with \$v\$](#) . The former is a set $\mathcal{I}_H$ of vertices such that $|\mathcal{I}_H| = 8$, such that $u \in \mathcal{I}_H$ and $v \notin \mathcal{I}_H$, such that $G[\mathcal{I}_H]$ has degree at most 3, and such that the degree of u in $G[\mathcal{I}_H]$ is at most 1. Indeed, the set $(\mathcal{I}_{H_2}^w \setminus \{w\}) \cup \{u, t_1, t_2\}$ satisfies these properties. The definition of a 8-over-11 set with v is analogous, with v in place of u and vice versa, and is realized by the set $\mathcal{I}_{H_2}^v \cup \{t_1, t_2\}$.

- Suppose that $h_1 = 5$ and $h_2 = 7$, the case in which $h_1 = 7$ and $h_2 = 5$ is symmetric. By assumption, H_2 admits a 5-over-7 set with w , say $\mathcal{I}_{H_2}^w$, and a 5-over-7 set with v , say $\mathcal{I}_{H_2}^v$.

Suppose first that w is adjacent to at most 2 out of the 3 vertices t_1, t_2 , and t_3 of H_1 different from u and w . Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t_1, t_2, t_3\}$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{8}{11} > \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 5-over-7 set with w and since w is the only vertex of H_2 adjacent to t_1, t_2 or t_3 , we have

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 1 in $G[\mathcal{I}_{H_2}^w]$, hence at most 3 in $G[\mathcal{I}_H]$, by the assumption that it is not adjacent to all of t_1, t_2 , and t_3 . Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

Suppose next that w is adjacent to all of t_1, t_2 , and t_3 . This implies that u is a neighbor of at most one of t_1, t_2 , and t_3 . It follows that H admits a 8-over-11 set with u and a 8-over-11 set with v . The former is realized by the set $(\mathcal{I}_{H_2}^w \setminus \{w\}) \cup \{u, t_1, t_2, t_3\}$ and the latter by the set $\mathcal{I}_{H_2}^v \cup \{t_1, t_2, t_3\}$.

Although we cannot exclude the existence of a uv -split subgraph with 11 vertices, we will use the existence of a 8-over-11 set with u and of a 8-over-11 set with v in later cases.

Case h=12: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 12$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 6, 9, or 10 vertices, we have that either one of h_1 and h_2 is 2 and the other one is 11, or one of h_1 and h_2 is 5 and the other one is 8.

- Suppose that $h_1 = 2$ and $h_2 = 11$, the case in which $h_1 = 11$ and $h_2 = 2$ is symmetric. By assumption, H_2 admits a 8-over-11 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{8}{12} = \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 8-over-11 set with w , we have that $G[\mathcal{I}_H]$ has degree at most 3. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.
- Suppose that $h_1 = 5$ and $h_2 = 8$, the case in which $h_1 = 8$ and $h_2 = 5$ is symmetric. By assumption, H_2 admits a 6-over-8 set with w , say $\mathcal{I}_{H_2}^w$. Let $\mathcal{I}_{H_2}$ be the set $\mathcal{I}_{H_2}^w \setminus \{w\}$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2} \cup \{t_1, t_2, t_3\}$, where t_1, t_2 , and t_3 are the vertices of H_1 different from u and w . Note that $|\mathcal{I}_{H_2}| = 5$, hence $|\mathcal{I}_H|/|V(H)| = \frac{8}{12} = \frac{2}{3}$. Since u and w are the only vertices of H_1 with neighbors in H_2 , each of t_1, t_2 , and t_3 has degree at most 2 in $G[\mathcal{I}_H]$. Since $\mathcal{I}_{H_2}^w$ is a 6-over-8 set, we have that $G[\mathcal{I}_{H_2}^w]$ and then also $G[\mathcal{I}_{H_2}]$ has degree at most 3. Since v and w are the only vertices of H_2 with neighbors in H_1 , it follows that every vertex in $\mathcal{I}_{H_2}$ has degree at most 3 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 12 vertices.

Case h=13: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 13$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 6, 9, 10, or 12 vertices, we have that either one of h_1 and h_2 is 3 and the other one is 11, or $h_1 = h_2 = 7$.

- Suppose that $h_1 = 3$ and $h_2 = 11$, the case in which $h_1 = 11$ and $h_2 = 3$ is symmetric. By assumption, H_2 admits a 8-over-11 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t_1\}$, where t_1 is the vertex of H_1 different from u and w . Note that $|\mathcal{I}_H|/|V(H)| = \frac{9}{13} > \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 8-over-11 set with w , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has at most one neighbor in $\mathcal{I}_{H_2}^w$, hence it has degree at most 2 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.
- Suppose that $h_1 = h_2 = 7$. By assumption, H_1 admits a 5-over-7 set with w , say $\mathcal{I}_{H_1}^w$, and H_2 admits a 5-over-7 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_1}^w \cup \mathcal{I}_{H_2}^w$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{9}{13} > \frac{2}{3}$. Since $\mathcal{I}_{H_1}^w$ and $\mathcal{I}_{H_2}^w$ are 5-over-7 sets with w , since u and w are the only vertices of H_1 with neighbors in H_2 , and since v and w are the only vertices of H_2 with neighbors in H_1 , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 1 in each of $G[\mathcal{I}_{H_1}^w]$ and $G[\mathcal{I}_{H_2}^w]$, hence at most 2 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 13 vertices.

Case h=14: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 14$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 6, 9, 10, 12, or 13 vertices, we have that either one of h_1 and h_2 is 4 and the other one is 11, or one of h_1 and h_2 is 7 and the other one is 8.

- Suppose that $h_1 = 4$ and $h_2 = 11$, the case in which $h_1 = 11$ and $h_2 = 4$ is symmetric. By assumption, H_2 admits a 8-over-11 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_2}^w \cup \{t_1, t_2\}$, where t_1 and t_2 are the vertices of H_1 different from u and w . Note that $|\mathcal{I}_H|/|V(H)| = \frac{10}{14} > \frac{2}{3}$. Since

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

$\mathcal{I}_{H_2}^w$ is a 8-over-11 set with w , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has at most one neighbor in $\mathcal{I}_{H_2}^w$, hence it has degree at most 3 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

- Suppose that $h_1 = 7$ and $h_2 = 8$, the other case is symmetric. By assumption, H_1 admits a 5-over-7 set with w , say $\mathcal{I}_{H_1}^w$, and H_2 admits a 6-over-8 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_1}^w \cup \mathcal{I}_{H_2}^w$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{10}{14} > \frac{2}{3}$. Since $\mathcal{I}_{H_1}^w$ is a 5-over-7 set with w , since $\mathcal{I}_{H_2}^w$ is a 6-over-8 set with w , since u and w are the only vertices of H_1 with neighbors in H_2 , and since v and w are the only vertices of H_2 with neighbors in H_1 , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 1 in $G[\mathcal{I}_{H_1}^w]$ and at most 2 in $G[\mathcal{I}_{H_2}^w]$, hence at most 3 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 14 vertices.

Case h=15: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 15$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 6, 9, 10, 12, 13, or 14 vertices, we have that either one of h_1 and h_2 is 5 and the other one is 11, or $h_1 = h_2 = 8$.

- Suppose that $h_1 = 5$ and $h_2 = 11$, the case in which $h_1 = 11$ and $h_2 = 5$ is symmetric. By assumption, H_2 admits a 8-over-11 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := (\mathcal{I}_{H_2}^w \setminus \{w\}) \cup \{t_1, t_2, t_3\}$, where t_1, t_2 , and t_3 are the vertices of H_1 different from u and w . Note that $|\mathcal{I}_H|/|V(H)| = \frac{10}{15} = \frac{2}{3}$. Since $\mathcal{I}_{H_2}^w$ is a 8-over-11 set with w , we have that $G[\mathcal{I}_H]$ has degree at most 3. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.
- Suppose that $h_1 = h_2 = 8$. By assumption, H_1 admits a 6-over-8 set with w , say $\mathcal{I}_{H_1}^w$, and H_2 admits a 6-over-8 set with w , say $\mathcal{I}_{H_2}^w$. Let $\mathcal{I}_{H_1}$ be the set $\mathcal{I}_{H_1}^w \setminus \{w\}$ and let $\mathcal{I}_{H_2}$ be the set $\mathcal{I}_{H_2}^w \setminus \{w\}$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_1} \cup \mathcal{I}_{H_2}$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{10}{15} = \frac{2}{3}$. Since u and w are the only vertices of H_1 with neighbors in H_2 , the degree in $G[\mathcal{I}_H]$ of each vertex in $\mathcal{I}_{H_1}$ is the same as its degree in $G[\mathcal{I}_{H_1}]$, hence at most

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

3. Analogously, the degree in $G[\mathcal{I}_H]$ of each vertex in $\mathcal{I}_{H_2}$ is at most 3. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired.

We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 15 vertices.

Case h=16: We observe that, with the current assumptions, G does not contain any edge uv such that a uv -split subgraph H of G has $h = 16$ vertices. Indeed, since $h_1 + h_2 = h + 1$, if H had $h = 16$ vertices, one of H_1 and H_2 would contain 6, 9, 10, 12, 13, 14, or 15 vertices, which we already excluded.

Case h=17: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 17$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 9, 10, 12, 13, 14, 15, or 16 vertices, we have that one of h_1 and h_2 is 7 and the other one is 11. Suppose that $h_1 = 7$ and $h_2 = 11$, as the other case is symmetric. By assumption, H_1 admits a 5-over-7 set with w , say $\mathcal{I}_{H_1}^w$, and H_2 admits a 8-over-11 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_1}^w \cup \mathcal{I}_{H_2}^w$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{12}{17} > \frac{2}{3}$. Since $\mathcal{I}_{H_1}^w$ is a 5-over-7 set with w , since $\mathcal{I}_{H_2}^w$ is a 8-over-11 set with w , since u and w are the only vertices of H_1 with neighbors in H_2 , and since v and w are the only vertices of H_2 with neighbors in H_1 , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 1 in each of $G[\mathcal{I}_{H_1}^w]$ and $G[\mathcal{I}_{H_2}^w]$, hence at most 2 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired. We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 17 vertices.

Case h=18: Suppose that G contains an edge uv such that a uv -split subgraph H of G has $h = 18$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has 10, 12, 13, 14, 15, 16, or 17 vertices, we have that one of h_1 and h_2 is 8 and the other one is 11. Suppose that $h_1 = 8$ and $h_2 = 11$, as the other case is symmetric. By assumption, H_1 admits a 6-over-8 set with w , say $\mathcal{I}_{H_1}^w$, and H_2 admits a 8-over-11 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_1}^w \cup \mathcal{I}_{H_2}^w$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{13}{18} > \frac{2}{3}$. Since $\mathcal{I}_{H_1}^w$ is a 6-over-8 set with w , since $\mathcal{I}_{H_2}^w$ is a 8-over-11 set with w , since u and w are the only vertices of H_1 with neighbors in H_2 , and since v and w are the only vertices of H_2 with neighbors in H_1 , we have that $G[\mathcal{I}_H]$ has degree at most 3; in particular, w has degree at most 2 in $G[\mathcal{I}_{H_1}^w]$ and at most 1 in $G[\mathcal{I}_{H_2}^w]$, hence at most 3 in $G[\mathcal{I}_H]$. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3,

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

as desired. We henceforth assume that G has no edge uv such that a uv -split subgraph of G has 18 vertices.

Case $h=19$: We observe that, with the current assumptions, G does not contain any edge uv such that a uv -split subgraph H of G has $h = 19$ vertices. Indeed, since $h_1 + h_2 = h + 1$, if H had $h = 19$ vertices, one of H_1 and H_2 would contain 9, 10, 12, 13, 14, 15, 16, 17, or 18 vertices, which we already excluded.

Case $h=20$: We observe that, with the current assumptions, G does not contain any edge uv such that a uv -split subgraph H of G has $h = 20$ vertices. Indeed, since $h_1 + h_2 = h + 1$, if H had $h = 20$ vertices, one of H_1 and H_2 would contain 10, 12, 13, 14, 15, 16, 17, 18, or 19 vertices, which we already excluded.

Case $h=21$: By [Lemma 7.3](#), applied with $\ell = 10$, since G has no edge uv such that a uv -split subgraph of G has a number of vertices between 12 and 20, we have that G has an edge uv such that a uv -split subgraph has $h = 21$ vertices. Since $h_1 + h_2 = h + 1$ and G has no edge xy such that an xy -split subgraph of G has a number of vertices between 12 and 20, we have that h_1 and h_2 are both 11. By assumption, H_1 admits a 8-over-11 set with w , say $\mathcal{I}_{H_1}^w$, and H_2 admits a 8-over-11 set with w , say $\mathcal{I}_{H_2}^w$. Then we can define $\mathcal{I}_H := \mathcal{I}_{H_1}^w \cup \mathcal{I}_{H_2}^w$. Note that $|\mathcal{I}_H|/|V(H)| = \frac{15}{21} > \frac{2}{3}$. Since $\mathcal{I}_{H_1}^w$ and $\mathcal{I}_{H_2}^w$ are 8-over-11 sets with w , it follows that the degree of w in each of $G[\mathcal{I}_{H_1}^w]$ and $G[\mathcal{I}_{H_2}^w]$ is at most 1, hence its degree in $G[\mathcal{I}_H]$ is at most 2. Since the degree in $G[\mathcal{I}_H]$ of every vertex in $\mathcal{I}_{H_i}^w \setminus \{w\}$ is the same as its degree in $G[\mathcal{I}_{H_i}^w]$, for $i = 1, 2$, it follows that the degree of $G[\mathcal{I}_H]$ is at most 3. Since u and v do not belong to $\mathcal{I}_H$, by [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2}{3}n$ vertices and $G[\mathcal{I}]$ has degree at most 3, as desired. $\square$

We conclude with the theorem for induced subgraphs with degree at most $k \geq 4$.

Theorem 7.6. *For every $n \geq 1$, for every n -vertex outerplanar graph G , and for every integer $k \geq 4$, there exists a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k and $|\mathcal{I}| \geq \frac{2k+1}{2k+5}n$ (if k is even) or $|\mathcal{I}| \geq \frac{2k+2/3}{2k+14/3}n$ (if k is odd).*

Proof. Let G be an n -vertex maximal outerplanar graph and let k be an integer greater than or equal to 4. We assume first that k is even, we will discuss later the case in which k is odd.

The proof is by induction on n . In the base case we have $n \leq \frac{3k}{2} + 3$. If $n \leq k + 1$, then we can set $\mathcal{I} := V(G)$ and obviously $G[\mathcal{I}] = G$ has degree at most k and $|\mathcal{I}| > \frac{2k+1}{2k+5}n$. If $k + 2 \leq n \leq \frac{3k}{2} + 3$, then the proof is similar to

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

the one for the base case of [Theorem 7.5](#). Namely, we can initialize $\mathcal{I} = V(G)$ and then repeatedly remove from $\mathcal{I}$ a vertex with degree at least $k + 1$ in $G[\mathcal{I}]$, while such a vertex exists. The removal of the first vertex from $\mathcal{I}$ removes at least k internal edges from $G[\mathcal{I}]$ (by the same arguments presented for the case $n = 11$ in the proof of [Theorem 7.5](#)), while the removal of the second vertex from $\mathcal{I}$ removes at least $k - 1$ internal edges from $G[\mathcal{I}]$. Since G has $n - 3 \leq \frac{3k}{2}$ internal edges, and since $2k - 1 > \frac{3k}{2}$ for $k > 2$, it follows that a single vertex is removed from $\mathcal{I}$. Hence, $|\mathcal{I}| = n - 1 \geq \frac{2k+1}{2k+5}n$, where the last inequality is equivalent to $n \geq \frac{2k+5}{4}$, which holds true by the assumption $n \geq k + 2$ and since $k + 2 \geq \frac{2k+5}{4}$ for $k \geq 0$.

In the inductive case, we have $n \geq \frac{3k}{2} + 4$. As in [Theorem 7.5](#), the general strategy consists of defining a uv -split subgraph H of G with a suitable size and of then selecting a large set $\mathcal{I}_H \subseteq V(H)$ of vertices, with $u, v \notin \mathcal{I}_H$, such that $G[\mathcal{I}_H]$ has degree at most k . Applying induction on $\overline{H}$ to get a set $\mathcal{I}_{\overline{H}}$ and then applying [Lemma 7.4](#) allows us to obtain the desired set $\mathcal{I}$.

Let H be a uv -split subgraph of G with h vertices. A set $\mathcal{I}_H \subseteq V(H)$ is called *good* for H if it contains neither u nor v , if $G[\mathcal{I}_H]$ has degree at most k , and if:

- $k + 3 \leq h \leq \frac{3k}{2} + 3$ and $|\mathcal{I}_H| \geq h - 2$; or
- $\frac{3k}{2} + 4 \leq h \leq 2k + 4$ and $|\mathcal{I}_H| \geq h - 3$; or
- $2k + 5 \leq h \leq 3k + 5$ and $|\mathcal{I}_H| \geq h - 4$.

Also, a set $\mathcal{I}_H \subseteq V(H)$ is an *$(h - 2)$ -over- h set with u* if $|\mathcal{I}_H| = h - 2$, if $u \in \mathcal{I}_H$ and $v \notin \mathcal{I}_H$, if $G[\mathcal{I}_H]$ has degree at most k , and if the degree of u in $G[\mathcal{I}_H]$ is at most $h - k - 3$. An *$(h - 2)$ -over- h set with v* is defined analogously, with v in place of u and vice versa. Finally, a set $\mathcal{I}_H \subseteq V(H)$ is an *$(h - 1)$ -over- h set with u and v* if $|\mathcal{I}_H| = h - 1$, if $u, v \in \mathcal{I}_H$, if $G[\mathcal{I}_H]$ has degree at most k , and if the degree of u and v in $G[\mathcal{I}_H]$ is at most $h - k - 2$.

We are going to prove the following three statements.

- (S1) If $h = k + 3$, then there exists a good set for H .
- (S2) Suppose that:
 - $k + 4 \leq h \leq \frac{3k}{2} + 3$;
 - G has no st -split subgraph with $k + 3$ vertices; and

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

- for every integer j with $k + 4 \leq j \leq h - 1$ and for every st -split subgraph J of G with j vertices, we have that J has a $(j - 2)$ -over- j set with s , a $(j - 2)$ -over- j set with t , and a $(j - 1)$ -over- j set with s and t .

Then there exists a good set for H , or there exist an $(h - 2)$ -over- h set with u , an $(h - 2)$ -over- h set with v , and an $(h - 1)$ -over- h set with u and v .

- (S3) Suppose that:

- $\frac{3k}{2} + 4 \leq h \leq 3k + 5$;
- G has no st -split subgraph with $k + 3$ vertices;
- for every integer j with $k + 4 \leq j \leq \frac{3k}{2} + 3$ and for every st -split subgraph J of G with j vertices, we have that J has a $(j - 2)$ -over- j set with s , a $(j - 2)$ -over- j set with t , and a $(j - 1)$ -over- j set with s and t ; and
- for every integer j with $\frac{3k}{2} + 4 \leq j \leq h - 1$, G has no st -split subgraph with j vertices.

Then there exists a good set for H .

Before proving the statements, we show that they imply the theorem (if k is even).

First, suppose that G has a uv -split subgraph H with $k + 3$ vertices. By statement (S1), there exists a good set $\mathcal{I}_H$ for H ; by definition, $|\mathcal{I}_H| = k + 1$. Hence, $\frac{|\mathcal{I}_H|}{|V(H)|} = \frac{k+1}{k+3} > \frac{2k+1}{2k+5}$, where the last inequality is true for every value of k . Also, $\mathcal{I}_H$ contains $k + 1$ vertices, hence $G[\mathcal{I}_H]$ has degree at most k . By [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2k+1}{2k+5}n$ vertices and $G[\mathcal{I}]$ has degree at most k , as desired. We henceforth assume that G has no edge uv such that a uv -split subgraph of G has $k + 3$ vertices.

Second, suppose that G has a uv -split subgraph H with h vertices, where $k + 4 \leq h \leq \frac{3k}{2} + 3$, such that there exists no $(h - 2)$ -over- h set with u , or there exists no $(h - 2)$ -over- h set with v , or there exists no $(h - 1)$ -over- h set with u and v . Among all such subgraphs, choose one such that h is minimum. Then the assumptions of statement (S2) are met, thus there exists a good set $\mathcal{I}_H$ for H . Hence, $\frac{|\mathcal{I}_H|}{|V(H)|} = \frac{h-2}{h} > \frac{k+1}{k+3} > \frac{2k+1}{2k+5}$, where the first inequality comes from $h \geq k + 4$. Also, $G[\mathcal{I}_H]$ has degree at most k since $\mathcal{I}_H$ is a good set. By [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2k+1}{2k+5}n$ vertices and $G[\mathcal{I}]$

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

has degree at most k , as desired. We henceforth assume that for every integer h with $k + 4 \leq h \leq \frac{3k}{2} + 3$ and for every uv -split subgraph H of G with h vertices, H has an $(h - 2)$ -over- h set with u , an $(h - 2)$ -over- h set with t , and an $(h - 1)$ -over- h set with u and v .

Third, setting $\ell = \frac{3k}{2} + 2$ and observing that $n \geq \ell + 2$ by the hypothesis of the inductive case, [Lemma 7.3](#) ensures that G contains a uv -split subgraph H with a number h of vertices between $\frac{3k}{2} + 4$ and $3k + 5$. Among all such subgraphs, choose one such that h is minimum. Then the assumptions of statement (S3) are met, thus there exists a good set $\mathcal{I}_H$ for H . If $\frac{3k}{2} + 4 \leq h \leq 2k + 4$, then $\frac{|\mathcal{I}_H|}{|V(H)|} = \frac{h-3}{h} \geq \frac{(3k/2)+1}{(3k/2)+4} > \frac{2k+1}{2k+5}$, where the last inequality is true for $k \geq 0$, while if $2k + 5 \leq h \leq 3k + 5$, then $\frac{|\mathcal{I}_H|}{|V(H)|} = \frac{h-4}{h} \geq \frac{2k+1}{2k+5}$. Also, $G[\mathcal{I}_H]$ has degree at most k since $\mathcal{I}_H$ is a good set. By [Lemma 7.4](#), the set $\mathcal{I} := \mathcal{I}_H \cup \mathcal{I}_{\overline{H}}$ contains at least $\frac{2k+1}{2k+5}n$ vertices and $G[\mathcal{I}]$ has degree at most k , as desired. This completes the induction.

It remains to prove the three statements. We denote by w the vertex such that the cycle (u, v, w) bounds an internal face of $\mathcal{O}_H$, by H_1 the uw -split subgraph of G not containing v , and by H_2 the vw -split subgraph of G not containing u . We let h_1 and h_2 be the number of vertices of H_1 and H_2 , respectively, where $h_1 \geq 2$, $h_2 \geq 2$, and $h_1 + h_2 = h + 1$.

Statement (S1). We define $\mathcal{I}_H := V(H) \setminus \{u, v\}$. Obviously, $|\mathcal{I}_H| = k + 1 = h - 2$ and $G[\mathcal{I}_H]$ has degree at most k , given that $\mathcal{I}_H$ contains $k + 1$ vertices, hence $\mathcal{I}_H$ is a good set.

Statement (S2). We distinguish three cases.

- In the *first case*, vertex w has at least $k + 1$ neighbors in $V(H) \setminus \{u, v\}$. We prove that the set $\mathcal{I}_H^u := V(H) \setminus \{w, v\}$ is an $(h - 2)$ -over- h set with u . The proof exploits the fact that any two vertices of H share at most two neighbors in H (as otherwise such two vertices and their common neighbors would induce $K_{2,3}$, whereas H is an outerplanar graph).

First, $|\mathcal{I}_H^u| = h - 2$, as desired. Second, $u \in \mathcal{I}_H^u$ and $v \notin \mathcal{I}_H^u$, by construction. Third, we prove that any vertex $z \neq u$ of $\mathcal{I}_H^u$ has degree at most k in $G[\mathcal{I}_H^u]$. Namely, w has at least $k + 2$ neighbors in $\mathcal{I}_H^u$ (at least $k + 1$ in $V(H) \setminus \{u, v\}$, by hypothesis, plus u). One of such neighbors might be z and at most 2 of them might be shared with z . Hence, out of the $h - 3$ vertices of $\mathcal{I}_H^u$ different from z , at least $(k - 1)$ are different from z and are not neighbors of z in $G[\mathcal{I}_H^u]$, which proves that the degree of z in $G[\mathcal{I}_H^u]$ is at most $(h - 3) - (k - 1)$. Since $h \leq \frac{3k}{2} + 3$, we have $(h - 3) - (k - 1) \leq \frac{k}{2} + 1 \leq k$, where the last inequality is equivalent to

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

$k \geq 2$. Finally, we prove that u has degree at most $h - k - 3$ in $G[\mathcal{I}_H^u]$. Note that u and w share v as a neighbor in H , hence out of the at least $k + 2$ neighbors of w in $\mathcal{I}_H^u$, at most one of them is a neighbor of u . Hence, out of the $h - 3$ vertices of $\mathcal{I}_H^u$ different from u , at least k are different from u and are not neighbors of u in $G[\mathcal{I}_H^u]$, which proves that the degree of u in $G[\mathcal{I}_H^u]$ is at most $(h - 3) - k$. Observe that $h - k - 3 \leq k$ since $h \leq \frac{3k}{2} + 3$ and $k \geq 0$.

The proof that the set $\mathcal{I}_H^v := V(H) \setminus \{u, w\}$ is an $(h - 2)$ -over- h set with v is symmetric. The proof that the set $\mathcal{I}_H^{uv} := V(H) \setminus \{w\}$ is an $(h - 1)$ -over- h set with u and v is also analogous; in particular, now the degree of any vertex $z \notin \{u, v\}$ of $\mathcal{I}_H^{uv}$ in $G[\mathcal{I}_H^{uv}]$ might be as large as $(h - 2) - (k - 1) = h - k - 1$, that is, one more than in $G[\mathcal{I}_H^u]$. Since $h \leq \frac{3k}{2} + 3$, we have $h - k - 1 \leq \frac{k}{2} + 2 \leq k$, where the last inequality is equivalent to $k \geq 4$. Also, the degree of u in $\mathcal{I}_H^{uv}$ is one more than in $G[\mathcal{I}_H^u]$, since u is a neighbor of v , hence it is at most $(h - 2) - k$, as required.

- In the *second case*, vertex w has at most k neighbors in $V(H) \setminus \{u, v\}$, $h_1 \leq k + 2$, and $h_2 \leq k + 2$. We prove that the set $\mathcal{I}_H := V(H) \setminus \{u, v\}$ is a good set for H . First, $|\mathcal{I}_H| = h - 2$, as desired. Second, $u, v \notin \mathcal{I}_H$, by construction. Third, we prove that any vertex $z \in \mathcal{I}_H$ has degree at most k in $G[\mathcal{I}_H]$. If $z = w$, this comes from the assumption that w has at most k neighbors in $\mathcal{I}_H$. If $z \neq w$ and $z \in V(H_1)$ (the case in which $z \neq w$ and $z \in V(H_2)$ is symmetric), then the neighbors of z are all in $V(H_1)$. Since $h_1 \leq k + 2$ and since u and z are not neighbors of z , it follows that the degree of z in $G[\mathcal{I}_H]$ is at most k , as required.
- In the *third case*, vertex w has at most k neighbors in $V(H) \setminus \{u, v\}$, and $h_1 \geq k + 3$ or $h_2 \geq k + 3$. By the assumptions of statement (S2), $h_1 \neq k + 3$ and $h_2 \neq k + 3$, hence $h_1 \geq k + 4$ or $h_2 \geq k + 4$. Since $2k + 8 > \frac{3}{2}k + 3$, for every $k \geq 0$, it follows that $h_1 \geq k + 4$ and $h_2 \geq k + 4$ do not both hold. Assume, w.l.o.g., that $h_1 \geq k + 4$, as the case $h_2 \geq k + 4$ can be discussed symmetrically. Since $2k + 4 > \frac{3}{2}k + 3$, for every $k \geq 0$, we have $h_2 \leq k - 1$.

We prove that H has an $(h - 2)$ -over- h set with u . By the assumptions of statement (S2) and since $k + 4 \leq h_1 \leq h - 1$, we have that H_1 admits an $(h_1 - 1)$ -over- h_1 set with u and w , which we denote by $\mathcal{I}_{H_1}^{uw}$. We show that $\mathcal{I}_H^u := \mathcal{I}_{H_1}^{uw} \cup (V(H_2) \setminus \{v\})$ is an $(h - 2)$ -over- h set with u . First, $|\mathcal{I}_H^u| = h - 2$, given that $\mathcal{I}_{H_1}^{uw}$ contains all the vertices of H_1 , except for one,

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

and given that $V(H_2) \setminus \{v\}$ contains all the vertices of H_2 , except for v . Second, $u \in \mathcal{I}_H^u$, given that $u \in \mathcal{I}_{H_1}^{uw}$, and $v \notin \mathcal{I}_H^u$, by construction. Third, we prove that any vertex $z \neq u$ of $\mathcal{I}_H^u$ has degree at most k in $G[\mathcal{I}_H^u]$. If $z \in \mathcal{I}_{H_1}^{uw}$ with $z \neq w$, then all the neighbors of z are in $V(H_1)$, hence z has degree at most k in $G[\mathcal{I}_H^u]$ since it has degree at most k in $G[\mathcal{I}_{H_1}^{uw}]$. Also, if $z \in V(H_2)$ with $z \neq w$, then all the neighbors of z are in $V(H_2)$, hence z has degree at most $k - 3$ in $G[\mathcal{I}_H^u]$ since $V(H_2)$ contains at most $k - 1$ vertices and v is not in $\mathcal{I}_H^u$. If $z = w$, then z has at most $h_1 - k - 2$ neighbors in $\mathcal{I}_{H_1}^{uw}$ and at most $h_2 - 2$ neighbors in $V(H_2)$ (that is, all the vertices of $V(H_2)$ except for v and w itself). Thus, its degree is at most $h_1 - k - 2 + h_2 - 2 = h - k - 3 \leq (\frac{3k}{2} + 3) - k - 3 = \frac{k}{2} < k$. Finally, we prove that u has degree at most $h - k - 3$ in $G[\mathcal{I}_H^u]$. We have that all the neighbors of u are in $\mathcal{I}_{H_1}^{uw}$, given that w and v are the only vertices of H_2 that are neighbors of u , that w also belongs to $\mathcal{I}_{H_1}^{uw}$, and that v does not belong to $\mathcal{I}_H^u$. It follows that u has degree at most $h_1 - k - 2 \leq h - k - 3$ in $G[\mathcal{I}_H^u]$, as required.

We next prove that H has an $(h - 2)$ -over- h set with v . Let $\mathcal{I}_{H_1}^w$ be an $(h_1 - 2)$ -over- h_1 set with w for H_1 . We show that $\mathcal{I}_H^v := \mathcal{I}_{H_1}^w \cup V(H_2)$ is an $(h - 2)$ -over- h set with v . First, $|\mathcal{I}_H^v| = h - 2$, given that $\mathcal{I}_{H_1}^w$ contains all the vertices of H_1 , except for two. Second, $u \notin \mathcal{I}_H^v$, since $\mathcal{I}_{H_1}^w$ is an $(h_1 - 2)$ -over- h_1 set with w , and $v \in \mathcal{I}_H^v$, by construction. Third, we prove that any vertex $z \neq v$ of $\mathcal{I}_H^v$ has degree at most k in $G[\mathcal{I}_H^v]$. If $z \in \mathcal{I}_{H_1}^w$ with $z \neq w$, then all the neighbors of z are in $V(H_1)$, hence z has degree at most k in $G[\mathcal{I}_H^v]$ since it has degree at most k in $G[\mathcal{I}_{H_1}^w]$. Also, if $z \in V(H_2)$ with $z \neq w$, then all the neighbors of z are in $V(H_2)$, hence z has degree at most $k - 2$ in $G[\mathcal{I}_H^v]$ since $V(H_2)$ contains at most $k - 1$ vertices. If $z = w$, then z has at most $h_1 - k - 3$ neighbors in $\mathcal{I}_{H_1}^w$ and at most $h_2 - 1$ neighbors in $V(H_2)$ (that is, all the vertices of $V(H_2)$ except for w itself). Thus, its degree is at most $h_1 - k - 3 + h_2 - 1 = h - k - 3 \leq (\frac{3k}{2} + 3) - k - 3 = \frac{k}{2} < k$. Finally, we prove that v has degree at most $h - k - 3$ in $G[\mathcal{I}_H^v]$. All the neighbors of v are in $V(H_2)$, given that u and w are the only vertices of H_1 that are neighbors of v , given that w also belongs to H_2 , and given that u does not belong to $\mathcal{I}_H^v$. It follows that v has degree at most $h_2 - 1 = h - h_1 \leq h - k - 4$ in $G[\mathcal{I}_H^v]$, as required.

We now prove that H has an $(h - 1)$ -over- h set with u and v . Let $\mathcal{I}_{H_1}^{uv}$ be an $(h_1 - 1)$ -over- h_1 set with u and w for H_1 . We show that $\mathcal{I}_H^{uv} := \mathcal{I}_{H_1}^{uv} \cup V(H_2)$ is an $(h - 1)$ -over- h set with u and v . First, $|\mathcal{I}_H^{uv}| = h - 1$, given that

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

$\mathcal{I}_{H_1}^{uw}$ contains all the vertices of H_1 , except for one. Second, $u \in \mathcal{I}_H^{uv}$ and $v \in \mathcal{I}_H^{uv}$, by construction. Third, we prove that any vertex $z \neq u, v$ of $\mathcal{I}_H^{uv}$ has degree at most k in $G[\mathcal{I}_H^{uv}]$. The arguments for the case in which $z \in \mathcal{I}_{H_1}^{uw}$ (in which $z \in V(H_2)$) with $z \neq w$ are the same as for the proof that $\mathcal{I}_H^u$ is an $(h-2)$ -over- h set with u (that $\mathcal{I}_H^v$ is an $(h-2)$ -over- h set with v , respectively). If $z = w$, then z has at most $h_1 - k - 2$ neighbors in $\mathcal{I}_{H_1}^{uw}$ and at most $h_2 - 1$ neighbors in $V(H_2)$ (that is, all the vertices of $V(H_2)$ except for w itself). Thus, its degree is at most $h_1 - k - 2 + h_2 - 1 = h - k - 2 \leq (\frac{3k}{2} + 3) - k - 2 = \frac{k}{2} + 1 < k$, where the last inequality holds true since $k \geq 4$. Finally, we prove that u and v have degree at most $h - k - 2$ in $G[\mathcal{I}_H^{uv}]$. We have that all the neighbors of u are in $\mathcal{I}_{H_1}^{uw}$, except for v . It follows that u has degree at most $(h_1 - k - 2) + 1 \leq h - k - 2$ in $G[\mathcal{I}_H^{uv}]$, as required. Also, all the neighbors of v are in $V(H_2)$, except for u . It follows that v has degree at most $(h_2 - 1) + 1 = (h - h_1) + 1 \leq h - k - 3$ in $G[\mathcal{I}_H^{uv}]$, as required.

Statement (S3). We distinguish four cases.

- In the *first case*, we have $h_1 \leq k + 2$ and $h_2 \leq k + 2$. We prove that the set $\mathcal{I}_H := V(H) \setminus \{u, w, v\}$ is a good set for H . First, $|\mathcal{I}_H| = h - 3$, which is the desired number of vertices, given that $h \geq k + 4$, by the assumptions of statement (S3). Second, $u, v \notin \mathcal{I}_H$, by construction. Third, we prove that any vertex $z \in \mathcal{I}_H$ has degree at most k in $G[\mathcal{I}_H]$. If $z \in V(H_1)$ (the case in which $z \in V(H_2)$ is symmetric), then the neighbors of z are all in $V(H_1)$. Since $h_1 \leq k + 2$ and since u, w , and z are either not in $\mathcal{I}_H$ or are not neighbors of z , it follows that the degree of z in $G[\mathcal{I}_H]$ is at most $k - 1$, as required.
- In the *second case*, we have that $h \leq 2k + 4$, $\max\{h_1, h_2\} > k + 2$, and $\min\{h_1, h_2\} \leq k + 2$. Assume that $h_1 > k + 2$ and that $h_2 \leq k + 2$, as the other case is analogous. By the assumptions of statement (S3), we have $k + 4 \leq h_1 \leq \frac{3k}{2} + 3$. Let $\mathcal{I}_{H_1}^w$ be an $(h_1 - 2)$ -over- h_1 set with w for H_1 . We show that $\mathcal{I}_H := \mathcal{I}_{H_1}^w \cup (V(H_2) \setminus \{v\})$ is a good set for H . First, $|\mathcal{I}_H| = h - 3$, which is equal to the desired number of vertices, given that $h \geq \frac{3k}{2} + 4$, by the assumptions of statement (S3). Second, $u \notin \mathcal{I}_H$, since $\mathcal{I}_{H_1}^w$ is an $(h_1 - 2)$ -over- h_1 set with w , and $v \notin \mathcal{I}_H$, by construction. Third, we prove that any vertex $z \in \mathcal{I}_H$ has degree at most k in $G[\mathcal{I}_H]$. If $z \in \mathcal{I}_{H_1}^w$ with $z \neq w$, then all the neighbors of z are in $V(H_1)$, hence z has degree at most k in $G[\mathcal{I}_H]$ since it has degree at most k in $G[\mathcal{I}_{H_1}^w]$. Also,

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

- if $z \in V(H_2)$ with $z \neq w$, then all the neighbors of z are in $V(H_2)$, hence z has degree at most k in $G[\mathcal{I}_H]$ since $V(H_2)$ contains at most $k+2$ vertices, v is not in $\mathcal{I}_H$ and z is not a neighbor of itself. Finally, w has at most $h_1 - k - 3$ neighbors in $\mathcal{I}_{H_1}^w$ and at most $h_2 - 2$ neighbors in $V(H_2) \setminus \{v\}$, hence its degree in $G[\mathcal{I}_H]$ is at most $h_1 - k - 3 + h_2 - 2 = h - k - 4 \leq k$, where the last inequality uses $h \leq 2k + 4$.
- In the *third case*, we have that $h \geq 2k + 5$, $\max\{h_1, h_2\} > k + 2$, and $\min\{h_1, h_2\} \leq k + 2$. Assume that $h_1 > k + 2$ and that $h_2 \leq k + 2$, as the other case is analogous. By the assumptions of statement (S3), we have $k + 4 \leq h_1 \leq \frac{3k}{2} + 3$. Let $\mathcal{I}_{H_1}^w$ be an $(h_1 - 2)$ -over- h_1 set with w for H_1 . We show that $\mathcal{I}_H := (\mathcal{I}_{H_1}^w \setminus \{w\}) \cup (V(H_2) \setminus \{w, v\})$ is a good set for H . First, $|\mathcal{I}_H| = h - 4$, which is equal to the desired number of vertices, given that $h \geq 2k + 5$, by the assumptions of statement (S3). Second, $u \notin \mathcal{I}_H$, since $\mathcal{I}_{H_1}^w$ is an $(h_1 - 2)$ -over- h_1 set with w , and $v \notin \mathcal{I}_H$, by construction. Third, we prove that any vertex $z \in \mathcal{I}_H$ has degree at most k in $G[\mathcal{I}_H]$. If $z \in \mathcal{I}_{H_1}^w$, then all the neighbors of z are in $V(H_1)$, hence z has degree at most k in $G[\mathcal{I}_H]$ since it has degree at most k in $G[\mathcal{I}_{H_1}^w]$. Also, if $z \in V(H_2)$, then all the neighbors of z are in $V(H_2)$, hence z has degree at most $k - 1$ in $G[\mathcal{I}_H]$ since $V(H_2)$ contains at most $k + 2$ vertices, w and v are not in $\mathcal{I}_H$, and z is not a neighbor of itself.
 - In the *fourth case*, we have $\min\{h_1, h_2\} > k + 2$. By the assumptions statement (S3), we have $k + 4 \leq h_1 \leq \frac{3k}{2} + 3$ and $k + 4 \leq h_2 \leq \frac{3k}{2} + 3$. Let $\mathcal{I}_{H_1}^w$ be an $(h_1 - 2)$ -over- h_1 set with w for H_1 and let $\mathcal{I}_{H_2}^w$ be an $(h_2 - 2)$ -over- h_2 set with w for H_2 . We show that $\mathcal{I}_H := \mathcal{I}_{H_1}^w \cup \mathcal{I}_{H_2}^w$ is a good set for H . First, $|\mathcal{I}_H| = h - 4$, given that $\mathcal{I}_{H_i}^w$ contains all the vertices of H_i , except for two, for $i = 1, 2$. Second, $u \notin \mathcal{I}_H$, since $\mathcal{I}_{H_1}^w$ is an $(h_1 - 2)$ -over- h_1 set with w , and $v \notin \mathcal{I}_H$, since $\mathcal{I}_{H_2}^w$ is an $(h_2 - 2)$ -over- h_2 set with w . Third, we prove that any vertex $z \in \mathcal{I}_H$ has degree at most k in $G[\mathcal{I}_H]$. If $z \in \mathcal{I}_{H_i}^w$ with $z \neq w$, for some $i \in \{1, 2\}$, then all the neighbors of z are in $V(H_i)$, hence z has degree at most k in $G[\mathcal{I}_H]$ since it has degree at most k in $G[\mathcal{I}_{H_i}^w]$. Also, w has degree at most $h_1 - k - 3$ in $G[\mathcal{I}_{H_1}^w]$ and at most $h_2 - k - 3$ in $G[\mathcal{I}_{H_2}^w]$, hence at most $h_1 + h_2 - 2k - 6 \leq h - 2k - 5 \leq k$, where we used $h \leq 3k + 5$.

This concludes the proof of the theorem for the case in which k is even.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

If k is odd, then the proof proceeds in the same way with the following differences (as a main numerical difference, the $\frac{3k}{2}$ terms for the even case become $\frac{3k-1}{2}$). First, in the base case we now have $n \leq \frac{3k-1}{2} + 3$; this case is handled with the same arguments as when k is even. Second, the definition of a good set $\mathcal{I}_H$ for H now requires $|\mathcal{I}_H| \geq h - 2$ if $k + 3 \leq h \leq \frac{3k-1}{2} + 3$, requires $|\mathcal{I}_H| \geq h - 3$ if $\frac{3k-1}{2} + 4 \leq h \leq 2k + 4$ and requires $|\mathcal{I}_H| \geq h - 4$ if $2k + 5 \leq h \leq 3k + 4$. Consequently, in statement (S2), the value of h satisfies $k + 4 \leq h \leq \frac{3k-1}{2} + 3$, while in statement (S3), the value of h satisfies $\frac{3k-1}{2} + 4 \leq h \leq 3k + 4$. Statements (S1)–(S3) then imply the statement of the theorem, by setting $\ell = \frac{3k-1}{2} + 2$. In particular, note that $\ell + 2 = \frac{3k-1}{2} + 4$ and $2\ell + 1 = 3k + 4$; these are the smallest and largest values of h in statement (S3). The reason why we need to settle for a smaller ratio, namely $\frac{2k+2/3}{2k+14/3}$ rather than $\frac{2k+1}{2k+5}$, when k is odd is that missing three vertices in a graph with at least $\frac{3k-1}{2} + 4$ vertices is not good enough for the ratio $\frac{2k+1}{2k+5}$, while it is for $\frac{2k+2/3}{2k+14/3}$. Indeed, $\frac{(3k-1)/2+1}{(3k-1)/2+4} = \frac{2k+2/3}{2k+14/3} < \frac{2k+1}{2k+5}$. The proof of the statements (S1)–(S3) proceeds almost verbatim with respect to the case in which k is even. In the proof of statement (S2), we used $h \leq \frac{3k}{2} + 3$ in order to prove the inequality $h - k - 1 \leq k$ (and the weaker inequalities $h - k - 3 \leq k$ and $h - k - 2 \leq k$). With $h \leq \frac{3k-1}{2} + 3$ rather than $h \leq \frac{3k}{2} + 3$ such inequality is obviously still satisfied, since the left term is smaller than before. Also, in the proof of statement (S3), we used $h \leq 3k + 5$ in order to prove that the degree of w , which is upper bounded by $h - 2k - 5$, is smaller than or equal to k . Again, since now $h \leq 3k + 5$, the degree of w is still smaller than or equal to k .

This concludes the proof of the theorem. $\square$

7.4 Induced Subgraphs of Planar Graphs

In this section, we consider the size of the largest induced subgraph of degree at most k that one is guaranteed to find in an n -vertex planar graph. [Table 7.3](#) shows the coefficients of the linear terms in the size of the induced subgraphs we can find, for various values of k .

	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
LB	> 0.384	> 0.457	0.5	> 0.571	0.625
UB	< 0.572	0.6	0.72	< 0.769	0.8

Table 7.3: Results on planar graphs with different values of k .

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

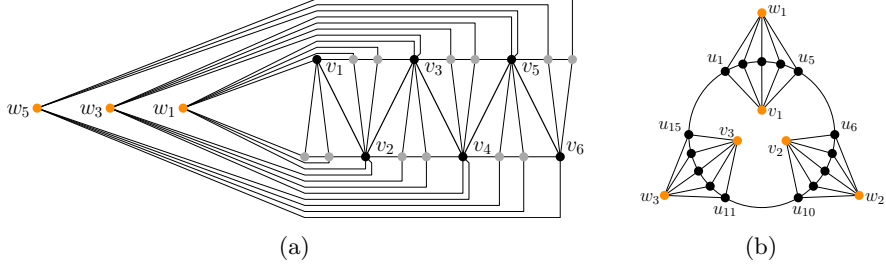


Figure 7.7: Illustration for **Theorem 7.7**. (a) The graph for the case $\mu = \frac{2k-2}{2k+1}n + 5$, with $n = 21$, $k = 3$, and $h = 3$. (b) The graph for the case $\mu = \frac{k+2}{k+4}n + 2$, $n = 21$, $k = 3$, and $h = 3$ (values of k for which $\frac{k+2}{k+4}n + 2 < \frac{2k-2}{2k+1}n + 5$ are too large to result in a readable figure).

We start by proving two upper bounds, the first one, $\frac{2k-2}{2k+1}n + O(1)$, is tighter for $k < 10$, while the second one, $\frac{k+2}{k+4}n + O(1)$, is tighter for $k > 10$.

Theorem 7.7. *For every $n \geq 1$ and for every integer $k \geq 3$, let $\mu = \min\{\frac{2k-2}{2k+1}n + 5, \frac{k+2}{k+4}n + 2\}$. There exists an n -vertex planar graph G with the following property. Consider any set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k . Then $|\mathcal{I}| \leq \mu$.*

Proof. We distinguish two cases, depending on which of $\frac{2k-2}{2k+1}n + 5$ and $\frac{k+2}{k+4}n + 2$ is smaller.

Suppose first that $\mu = \frac{2k-2}{2k+1}n + 5$. Let G be the n -vertex planar graph depicted in **Figure 7.7(a)** and defined as follows (note that G contains the graph from the proof of **Theorem 7.1** as a subgraph). Let $h = \lfloor \frac{n}{2k+1} \rfloor$. Then G contains a path $v_1, \dots, v_{2h}$, vertices $u_1^i, \dots, u_{k-1}^i$, for $i = 1, \dots, 2h$, and edges $u_j^i u_{j+1}^i$, $v_i u_j^i$, $v_{i-1} u_1^i$, and $u_{k-1}^i v_{i+1}$, whenever the end-vertices of such edges are in G . The graph also contains vertices $w_1, w_3, \dots, w_{2h-1}$, where w_{2i-1} is connected to v_{2i-2} (if $i > 1$), to v_{2i-1} , to v_{2i} , to v_{2i+1} (if $i < h$), and to $u_1^{2i-1}, \dots, u_{k-1}^{2i-1}, u_1^{2i}, \dots, u_{k-1}^{2i}$. Additional $n - h \cdot (2k + 1)$ isolated vertices, which do not play any role in the proof of the statement, are added to G , so that it has n vertices.

Consider any set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k . We prove that there exists a set $\mathcal{I}' \subseteq V(G)$ such that $G[\mathcal{I}']$ has degree at most k , such that $|\mathcal{I}'| = |\mathcal{I}|$, and such that $\mathcal{I}'$ does not contain any of $w_1, w_3, \dots, w_{2h-1}$. Suppose that $\mathcal{I}$ contains a vertex w_{2i-1} , for some $i \in \{1, \dots, h\}$. Then there exists a

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

vertex z among $u_1^{2i-1}, \dots, u_{k-1}^{2i-1}, u_1^{2i}, \dots, u_{k-1}^{2i}$, that is not in $\mathcal{I}$, as otherwise w_{2i-1} would have degree $2k-2$, which is larger than k since $k \geq 3$. Replace w_{2i-1} in $\mathcal{I}$ with z . Clearly, the size of $\mathcal{I}$ remains the same. Also, $G[\mathcal{I}]$ still has degree at most k . Indeed, suppose, for a contradiction, that a vertex z' has degree larger than k in $G[\mathcal{I}]$. Then z' is either z or a neighbor of z , given that z' either is not in $\mathcal{I}$ or is in $\mathcal{I}$ and has degree at most k before the replacement. If $z' = z$, it suffices to observe that z' has degree at most 4 in G , and one of its neighbors, namely w_{2i-1} , is not in $\mathcal{I}$ after the replacement. If z' is a neighbor of z , then z' is a vertex among v_{2i-2} (if $i > 1$), v_{2i-1} , v_{2i} , v_{2i+1} (if $i < h$), and $u_1^{2i-1}, \dots, u_{k-1}^{2i-1}, u_1^{2i}, \dots, u_{k-1}^{2i}$. However, all such vertices are neighbors of w_{2i-1} , hence the degree of z' in $G[\mathcal{I}]$ does not increase after the replacement. The repetition of this replacement results in the desired set $\mathcal{I}'$.

From the set $\mathcal{I}'$, a set $\mathcal{I}'' \subseteq V(G)$ such that $G[\mathcal{I}'']$ has degree at most k , such that $|\mathcal{I}''| = |\mathcal{I}'|$, and such that $\mathcal{I}''$ does not contain any of $w_1, w_3, \dots, w_{2h-1}, v_2, \dots, v_{2h-1}$ can be constructed as in the proof of [Theorem 7.1](#). Indeed, suppose that $\mathcal{I}'$ contains a vertex v_i , for some $i \in \{2, \dots, 2h-1\}$. Then there exists a vertex z among $u_{k-1}^{i-1}, u_1^i, u_2^i, \dots, u_{k-1}^i, u_1^{i+1}$ that does not belong to $\mathcal{I}'$, as otherwise v_i would have degree $k+1$ in $G[\mathcal{I}']$. Replace v_i in $\mathcal{I}'$ with z . Clearly, the size of $\mathcal{I}'$ remains the same. Also, $G[\mathcal{I}']$ still has degree at most k . Indeed, no vertex among $w_1, w_3, \dots, w_{2h-1}$ belongs to $\mathcal{I}'$. Also, every vertex u_j^ℓ has degree at most 4 in G and has one neighbor among $w_1, w_3, \dots, w_{2h-1}$, which is not in $\mathcal{I}'$, hence the degree of u_j^ℓ in $G[\mathcal{I}']$ is at most 3. Finally, z is a neighbor of v_i , which however is not in $\mathcal{I}'$ after the replacement, and might be a neighbor of v_{i-1} or v_{i+1} . However, v_{i-1} or v_{i+1} are also neighbors of v_i , hence if they are in $\mathcal{I}'$, their degree in $G[\mathcal{I}']$ does not increase. The repetition of this replacement results in the desired set $\mathcal{I}''$.

Note that $|\mathcal{I}''| \leq n - (2h-2) - h = n - 3\lfloor \frac{n}{2k+1} \rfloor + 2 \leq n - \frac{3n}{2k+1} + 5 = \frac{2k-2}{2k+1}n + 5$.

Suppose next that $\mu = \frac{k+2}{k+4}n + 2$. We define a planar graph G as follows, see [Figure 7.7\(b\)](#). Let $h = \lfloor \frac{n}{k+4} \rfloor$. Initialize G to a cycle with $n - 2h$ vertices, labeled $u_1, \dots, u_{n-2h}$. For $i = 1, \dots, h$, insert into G two vertices v_i and w_i , which are adjacent to the $k+2$ vertices u_j such $(i-1) \cdot (k+2) + 1 \leq j \leq i \cdot (k+2)$.

Consider any set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k . For $i = 1, \dots, h$, we have that, among the $k+4$ vertices $v_i, w_i, u_{(i-1) \cdot (k+2) + 1}, \dots, u_{i \cdot (k+2)}$, at least two do not belong to $\mathcal{I}$. Indeed, if at least $k+1$ among $u_{(i-1) \cdot (k+2) + 1}, \dots, u_{i \cdot (k+2)}$ belong to $\mathcal{I}$, then neither v_i nor w_i belongs to $\mathcal{I}$, as otherwise $G[\mathcal{I}]$ would contain a vertex of degree at least $k+1$. It follows that $|\mathcal{I}| \leq n - 2h = n - 2\lfloor \frac{n}{k+4} \rfloor \leq n - 2\frac{n}{k+4} + 2 = \frac{k+2}{k+4}n + 2$. $\square$

We now focus our attention on the lower bound. We present an algorithm,

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

called GREEDYREMOVAL, that given a (not necessarily planar) graph G and a target integer value $k \geq 3$, defines a large set of vertices $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k . As the name suggests, the algorithm follows a natural greedy strategy: It repeatedly removes from the current graph $\mathcal{G}$ (initially, we have $\mathcal{G} = G$) any vertex with maximum degree, until $\mathcal{G}$ has the target degree k . When this happens, the set of vertices of $\mathcal{G}$ is the desired set $\mathcal{I}$.

In order to get better bounds, we tweak this strategy a bit if $k = 3$ or $k = 4$. In these cases, once $\mathcal{G}$ has degree $k + 1$, we either repeatedly remove vertices of degree $k + 1$, or we act as follows. We augment $\mathcal{G}$ to a $(k + 1)$ -regular graph, in which we find a small dominating set $\mathcal{D}$. We remove from $\mathcal{G}$ the vertices in $\mathcal{D}$. Now $\mathcal{G}$ has degree k and its vertex set is the desired set $\mathcal{I}$. We choose the strategy that removes the smallest number of vertices. This algorithm is the basis for the following theorem.

Theorem 7.8. *For every $n \geq 1$, for every graph G with n vertices, m edges, and degree $d \geq 4$, and for every integer $3 \leq k \leq d - 1$, there exists a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k and such that:*

1. if $d = 4$ and $k = 3$, then $|\mathcal{I}| \geq \frac{7}{11}n - \frac{24}{11}$;
2. if $d = 5$ and $k = 3$, then $|\mathcal{I}| \geq \frac{35}{78}n - \frac{8}{13}$;
3. if $d \geq 6$ and $k = 3$, then $|\mathcal{I}| \geq \frac{10}{13}n - \frac{5}{39}m - \frac{8}{13}$;
4. if $d = 5$ and $k = 4$, then $|\mathcal{I}| \geq \frac{9}{14}n - \frac{1}{2}$;
5. if $d \geq 6$ and $k = 4$, then $|\mathcal{I}| \geq \frac{54}{59}n - \frac{9}{59}m - \frac{35}{59}$; and
6. if $d \geq 6$ and $k \geq 5$, then $|\mathcal{I}| \geq n - \frac{m}{k+1}$.

Proof. Before delving into the analysis of GREEDYREMOVAL, we introduce some notation. A simple, yet crucial, observation is that the degree of the current graph $\mathcal{G}$ does not increase throughout the algorithm execution. For any $i = d, d - 1, \dots, k$, we let $\mathcal{G}_i$ be the graph $\mathcal{G}$ the first time it has degree at most i ; thus, $\mathcal{G}_d = G$ and $\mathcal{G}_k$ is the graph whose vertex set is $\mathcal{I}$. It is possible that $\mathcal{G}_i = \mathcal{G}_{i-1}$; this happens if removing vertices of degree j from $\mathcal{G}_j$, for some $j \geq i + 1$, the degree of $\mathcal{G}$ drops to $i - 1$, or lower. We denote by r_i the number of vertices of degree i that are removed by GREEDYREMOVAL; these are the vertices that are in $\mathcal{G}_i$ and not in $\mathcal{G}_{i-1}$. Also, we denote by r_6^+ the sum $\sum_{j=6}^d r_j$, that is, the number of vertices of degree at least 6 that are removed by GREEDYREMOVAL.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

We are now ready to analyze GREEDYREMOVAL. In order to do so, we distinguish the six cases in the theorem's statement.

Case 1: $d = 4$ and $k = 3$. We show that $G = \mathcal{G}_4$ can be augmented to a 4-regular graph by adding at most 6 vertices, and several edges, to it. While there exist two vertices that are not adjacent and that both have degree smaller than 4, we add an edge between them. When no such an edge can be added, the vertices of degree smaller than 4 induce a clique K_x in $\mathcal{G}_4$, with $x \leq 4$. Then it suffices to add at most 6 vertices (and some further edges) to $\mathcal{G}_4$ in order to obtain a 4-regular graph $\mathcal{H}_4$, see Figure 7.8. Every h -vertex 4-regular graph

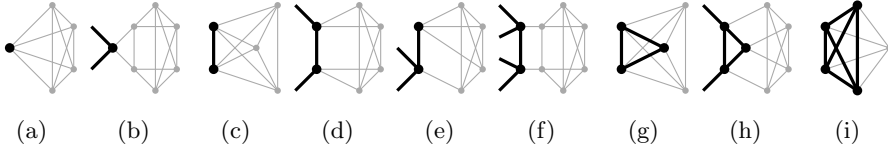


Figure 7.8: If the vertices of degree smaller than 4 induce a clique K_x in $\mathcal{G}_4$, it suffices to add at most 6 vertices (and some edges) to $\mathcal{G}_4$ in order to obtain a 4-regular graph $\mathcal{H}_4$. Vertices and edges in $\mathcal{G}_4$ are black, while vertices and edges in $\mathcal{H}_4$ and not in $\mathcal{G}_4$ are gray. Each vertex of K_x has degree at least $x - 1$ and at most 3 in $\mathcal{G}_4$. Also, the sum of the degrees of the vertices in K_x is even. This results in the following cases. (a) $x = 1$, one vertex of degree 0. (b) $x = 1$, one vertex of degree 2. (c) $x = 2$, two vertices of degree 1. (d) $x = 2$, two vertices of degree 2. (e) $x = 2$, one vertex of degree 1 and one vertex of degree 3. (f) $x = 2$, two vertices of degree 3. (g) $x = 3$, three vertices of degree 2. (h) $x = 3$, two vertices of degree 3 and one vertex of degree 2. (i) $x = 4$, four vertices of degree 3.

has a dominating set of size at most $\frac{4}{11}h$ [217], hence $\mathcal{H}_4$ has a dominating set $\mathcal{D}$ of size at most $\frac{4}{11}(n + 6)$. Let r_4 be the number of vertices of $\mathcal{G}_4$ that are in $\mathcal{D}$. Removing such vertices from $\mathcal{G}_4$ results in a graph $\mathcal{G}_3$ with degree at most 3. We thus get the following:

$$r_4 \leq \frac{4}{11}(n + 6). \quad (7.1)$$

The vertex set of $\mathcal{G}_3$ is our desired set $\mathcal{I}$, where $|\mathcal{I}| = n - r_4 \geq \frac{7}{11}n - \frac{24}{11}$.

Case 2: $d = 5$ and $k = 3$. The algorithm first removes r_5 vertices from G obtaining $\mathcal{G}_4$, and then removes r_4 vertices from $\mathcal{G}_4$ resulting in $\mathcal{G}_3$, whose vertex set is $\mathcal{I}$. We upper bound $r_4 + r_5$ in two different ways. First, by Equation (7.1) and since the number of vertices of $\mathcal{G}_4$ is $n - r_5$, we get:

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

$$r_4 + r_5 \leq \frac{4}{11}(n - r_5 + 6) + r_5 = \frac{4n + 7r_5 + 24}{11}. \quad (7.2)$$

Second, the number of vertices of degree 4 that are removed from $\mathcal{G}_4$ is at most $\frac{1}{4}$ times the number of edges of $\mathcal{G}_4$, since the removal of a degree-4 vertex removes 4 edges from $\mathcal{G}_4$. Also, the number of edges of $\mathcal{G}_4$ is equal to the number of edges of $\mathcal{G}_5$, which is at most $\frac{5n}{2}$ given that $\mathcal{G}_5$ has degree 5, minus 5 times the number of vertices of degree 5 that are removed from $\mathcal{G}_5$. Hence, we get:

$$r_4 + r_5 \leq \frac{\frac{5n}{2} - 5r_5}{4} + r_5 = \frac{5n - 2r_5}{8}. \quad (7.3)$$

We thus get:

$$r_4 + r_5 \leq \frac{43}{78}n + \frac{8}{13}. \quad (7.4)$$

Equation (7.4) comes from **Equation (7.2)** when $r_5 \leq \frac{23n}{78} - \frac{32}{13}$ and from **Equation (7.3)** when $r_5 \geq \frac{23n}{78} - \frac{32}{13}$. Therefore, it follows that $|\mathcal{I}| = n - r_4 - r_5 \geq \frac{35}{78}n - \frac{8}{13}$.

Case 3: $d \geq 6$ and $k = 3$. The algorithm first removes r_{6+} vertices from G to obtain $\mathcal{G}_5$, then removes r_5 vertices from $\mathcal{G}_5$ to obtain $\mathcal{G}_4$, and finally removes r_4 vertices from $\mathcal{G}_4$, resulting in $\mathcal{G}_3$, whose vertex set is $\mathcal{I}$. We upper bound $r_4 + r_5 + r_{6+}$ in three different ways. First, by **Equation (7.4)** and since the number of vertices of $\mathcal{G}_5$ is $n - r_{6+}$ we get:

$$r_4 + r_5 + r_{6+} \leq \frac{43}{78}(n - r_{6+}) + \frac{8}{13} + r_{6+} = \frac{43n + 35r_{6+} + 48}{78}. \quad (7.5)$$

Second, by **Equation (7.1)** and since the number of vertices of $\mathcal{G}_4$ is $n - r_5 - r_{6+}$, we get:

$$r_4 + r_5 + r_{6+} \leq \frac{4}{11}(n - r_5 - r_{6+} + 6) + r_5 + r_{6+} = \frac{4n + 7r_5 + 7r_{6+} + 24}{11}. \quad (7.6)$$

Third, as in Case 2, the number of vertices of degree 4 that are removed from $\mathcal{G}_4$ is at most $\frac{1}{4}$ times the number of edges of $\mathcal{G}_4$. Also, the number of edges of $\mathcal{G}_4$ is at most the number m of edges of G , minus 5 times the number of vertices of degree 5 that are removed from $\mathcal{G}_5$, minus 6 times the number of vertices of degree at least 6 that are removed from G . Hence, we get:

$$r_4 + r_5 + r_{6+} \leq \frac{m - 5r_5 - 6r_{6+}}{4} + r_5 + r_{6+} = \frac{m - r_5 - 2r_{6+}}{4}. \quad (7.7)$$

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

We thus get:

$$r_4 + r_5 + r_{6+} \leq \frac{3}{13}n + \frac{5}{39}m + \frac{8}{13}. \quad (7.8)$$

Equation (7.8) comes from Equation (7.5) when $r_{6+} \leq -\frac{5}{7}n + \frac{2}{7}m$, from Equation (7.6) when $r_5 + r_{6+} \leq -\frac{19}{91}n + \frac{55}{273}m - \frac{32}{13}$, and it comes from Equation (7.7) when $r_{6+} \geq -\frac{5}{7}n + \frac{2}{7}m$ and $r_5 + r_{6+} \geq -\frac{19}{91}n + \frac{55}{273}m - \frac{32}{13}$. The last implication is justified by observing that, for any fixed value of $r_5 + r_{6+}$, the quantity $\frac{m-r_5-2r_{6+}}{4}$ is maximized by choosing r_{6+} as small as possible, i.e., equal to $-\frac{5}{7}n + \frac{2}{7}m$. Then the value of r_5 which maximizes $\frac{m-r_5-2r_{6+}}{4}$ is $-\frac{19}{91}n + \frac{55}{273}m - \frac{32}{13} - (-\frac{5}{7}n + \frac{2}{7}m) = \frac{46}{91}n - \frac{23}{273}m - \frac{32}{13}$. It follows that $|\mathcal{I}| = n - r_4 - r_5 - r_{6+} \geq \frac{10}{13}n - \frac{5}{39}m - \frac{8}{13}$.

Case 4: $d = 5$ and $k = 4$. This case is similar to **Case 1**. Namely, we augment $G = \mathcal{G}_5$ to a 5-regular graph by adding at most 7 vertices, and several edges, to it. This is done by first adding edges between pairs of non-adjacent vertices whose degree is smaller than 5. When this is not possible, the vertices of degree smaller than 5 induce a clique K_x in $\mathcal{G}_5$. Then it suffices to add at most 7 vertices (and some further edges) to $\mathcal{G}_5$ in order to obtain a 5-regular graph $\mathcal{H}_5$, see Figure 7.9. Every h -vertex 5-regular graph has a dominating set of size at most $\frac{5}{14}h$ [218], hence $\mathcal{H}_5$ has a dominating set $\mathcal{D}$ of size at most $\frac{5}{14}(n+7)$. Let r_5 be the number of vertices of $\mathcal{G}_5$ that are in $\mathcal{D}$. Removing such vertices from $\mathcal{G}_5$ results in a graph $\mathcal{G}_4$ with degree at most 4. We thus get the following:

$$r_5 \leq \frac{5}{14}(n+7). \quad (7.9)$$

The vertex set of $\mathcal{G}_4$ is our desired set $\mathcal{I}$, where $|\mathcal{I}| = n - r_5 \geq \frac{9}{14}n - \frac{1}{2}$.

Case 5: $d \geq 6$ and $k = 4$. The algorithm first removes r_{6+} vertices from G obtaining $\mathcal{G}_5$, and then removes r_5 vertices from $\mathcal{G}_5$ resulting in $\mathcal{G}_4$, whose vertex set is $\mathcal{I}$. We upper bound $r_5 + r_{6+}$ in two different ways. First, by Equation (7.9) and since the number of vertices of $\mathcal{G}_5$ is $n - r_{6+}$, we get:

$$r_5 + r_{6+} \leq \frac{5}{14}(n - r_{6+} + 7) + r_{6+} = \frac{5n + 9r_{6+} + 35}{14}. \quad (7.10)$$

Second, the number of vertices of degree 5 that are removed from $\mathcal{G}_5$ is at most $\frac{1}{5}$ times the number of edges of $\mathcal{G}_5$, since the removal of a degree-5 vertex removes 5 edges from $\mathcal{G}_5$. Also, the number of edges of $\mathcal{G}_5$ is at most the number m of edges of G , minus 6 times the number of vertices of degree at least 6 that are removed from G . Hence, we get:

$$r_5 + r_{6+} \leq \frac{m - 6r_{6+}}{5} + r_{6+} = \frac{m - r_{6+}}{5}. \quad (7.11)$$

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

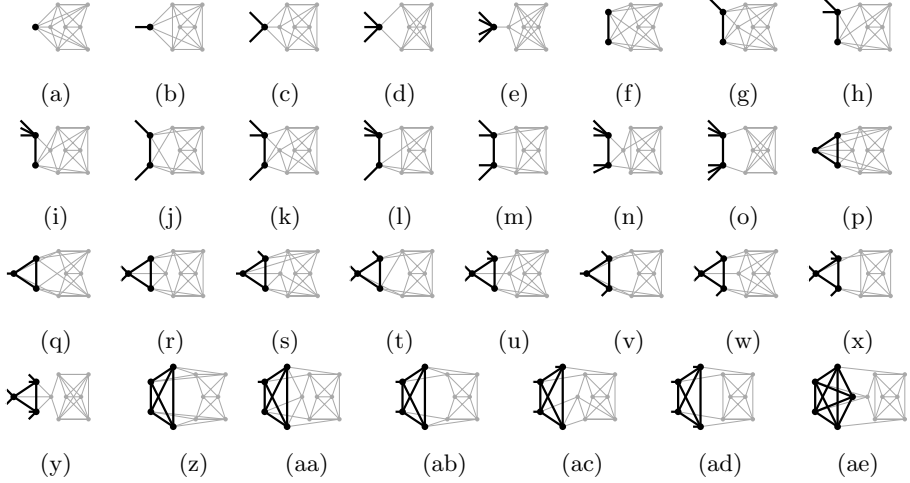


Figure 7.9: If the vertices of degree less than 5 induce a clique K_x in $\mathcal{G}_5$, it suffices to add at most 7 vertices (and some edges) to $\mathcal{G}_5$ to obtain a 5-regular graph $\mathcal{H}_5$. Vertices and edges in $\mathcal{G}_5$ are black, while those in $\mathcal{H}_5$ and not in $\mathcal{G}_5$ are gray. Each vertex of K_x has degree between $x - 1$ and 4 in $\mathcal{G}_5$, leading to the following cases. (a) $x = 1$, one deg-0 vertex. (b) $x = 1$, one deg-1 vertex. (c) $x = 1$, one deg-2 vertex. (d) $x = 1$, one deg-3 vertex. (e) $x = 1$, one deg-4 vertex. (f) $x = 2$, two deg-1 vertices. (g) $x = 2$, one deg-2 vertex and one deg-1 vertex. (h) $x = 2$, one deg-3 vertex and one deg-1 vertex. (i) $x = 2$, one deg-4 vertex and one deg-1 vertex. (j) $x = 2$, two deg-2 vertices. (k) $x = 2$, one deg-3 vertex and one deg-2 vertex. (l) $x = 2$, one deg-4 vertex and one deg-2 vertex. (m) $x = 2$, two deg-3 vertices. (n) $x = 2$, one deg-4 vertex and one deg-3 vertex. (o) $x = 2$, two deg-4 vertices. (p) $x = 3$, three deg-2 vertices. (q) $x = 3$, one deg-3 vertex and two deg-2 vertices. (r) $x = 3$, one deg-4 vertex and two deg-2 vertices. (s) $x = 3$, two deg-3 vertices and one deg-2 vertex. (t) $x = 3$, one deg-4 vertex, one deg-3 vertex, and one vertex of degree 2. (u) $x = 3$, two vertices of degree 4 and one deg-2 vertex. (v) $x = 3$, three deg-3 vertices. (w) $x = 3$, one deg-4 vertex and two deg-3 vertices. (x) $x = 3$, two deg-4 vertices and one deg-3 vertex. (y) $x = 3$, three deg-4 vertices. (z) $x = 4$, four deg-3 vertices. (aa) $x = 4$, one deg-4 vertex and three deg-3 vertices. (ab) $x = 4$, two deg-4 vertices and one deg-3 vertex. (ac) $x = 4$, three deg-4 vertices and three deg-3 vertices. (ad) $x = 4$, four deg-4 vertices. (ae) $x = 5$, five deg-4 vertices.

7. LARGE INDUCED SUBGRAPHS OF BOUNDED DEGREE IN OUTERPLANAR AND PLANAR GRAPHS

We thus get:

$$r_5 + r_{6+} \leq \frac{5}{59}n + \frac{9}{59}m + \frac{35}{59}. \quad (7.12)$$

Equation (7.12) comes from Equation (7.10) when $r_{6+} \leq -\frac{25}{59}n + \frac{14}{59}m - \frac{175}{59}$ and from Equation (7.11) when $r_{6+} \geq -\frac{25}{59}n + \frac{14}{59}m - \frac{175}{59}$. Therefore, it follows that $|\mathcal{I}| = n - r_5 - r_{6+} \geq \frac{54}{59}n - \frac{9}{59}m - \frac{35}{59}$.

Case 6: $d \geq 6$ and $k \geq 5$. The number of vertices of degree at least $k+1$ that are removed from G is at most $\frac{1}{k+1}$ times the number of edges of G , since the removal of a vertex with degree at least $k+1$ removes at least $k+1$ edges from G . Hence, we get:

$$\sum_{j=k+1}^d r_j \leq \frac{m}{k+1}. \quad (7.13)$$

Therefore, it follows that $|\mathcal{I}| = n - \sum_{j=k+1}^d r_j \geq n - \frac{m}{k+1}$. $\square$

Since a planar graph has at most $3n - 6$ edges, we get the following corollary.

Corollary 7.9. *For every $n \geq 1$, for every planar graph G with n vertices, and for every integer $k \geq 3$, there exists a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k and such that (i) if $k = 3$, then $|\mathcal{I}| \geq \frac{5}{13}n + \frac{2}{13}$; (ii) if $k = 4$, then $|\mathcal{I}| \geq \frac{27}{59}n + \frac{19}{59}$; and (iii) if $k \geq 5$, then $|\mathcal{I}| \geq \frac{k-2}{k+1}n + \frac{6}{k+1}$.*

Similarly, we can get better lower bounds for sparse families of planar graphs. For example, n -vertex planar graphs that admit a planar drawing in which every face has degree at least g (these include planar graphs of girth at least g) have at most $\frac{g(n-2)}{g-2}$ edges by Euler's formula, and hence have a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most k and such that $|\mathcal{I}| \geq \frac{(25g-60)n-14g+48}{39(g-2)}$, $|\mathcal{I}| \geq \frac{(45g-108)n-17g+70}{59(g-2)}$, and $|\mathcal{I}| \geq \frac{(gk-2k-2)n+2g}{(g-2)(k+1)}$ if $k = 3$, $k = 4$, or $k \geq 5$, respectively, by Theorem 7.8. As a notable implication, n -vertex bipartite planar graphs have a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ has degree at most 3 and such that $|\mathcal{I}| \geq \frac{20}{39}n - \frac{4}{39}$, which is strictly larger than the maximum size of an induced linear forest [181].

Large Induced Outerplanar Subgraphs in 2-Outerplanar Graphs

A *2-outerplanar graph* is a planar graph that admits a plane embedding such that, by removing the vertices incident to the outer face, one gets an outerplanar graph; the 2-outerplanar graph, equipped with such an embedding, is called *2-outerplane graph*. The class of 2-outerplanar graphs has been studied by Borradaile, Le and Sherman-Bennett [15] as an interesting benchmark for the maximum size of an induced forest or induced outerplane graph that one is guaranteed to find in a planar graph. They proved the best possible results for both problems: Every n -vertex 2-outerplane graph has a set of $n/2$ vertices that induces a forest (thus proving Albertson and Berman's conjecture for the 2-outerplanar graphs (see [Chapter 6](#))) and has a set of $2n/3$ vertices that induces an outerplane graph. Unfortunately, we point out a major flaw in their proof of the second result. What makes the flaw major is that the general strategy they use, that is, the set of vertices inducing an outerplane graph should contain all the vertices that are not incident to the outer face of the given 2-outerplane graph, plus, possibly, some additional vertices, cannot yield the desired bound. Indeed, we show that, for arbitrarily large values of n , there exists an n -vertex 2-outerplane graph such that any set of vertices that contains all the vertices that are not incident to the outer face and induces an outerplanar graph has at most $7n/11$ vertices. The aim of this chapter¹ is to

¹The content of this chapter is joint work with Fabrizio Frati and appeared as a technical report in [219]. At the time of this thesis defense, the work is under revision.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

recover the following result.

Theorem 8.1. *Let G be an n -vertex 2-outerplane graph. There exists a set $\mathcal{I} \subseteq V(G)$ with $|\mathcal{I}| \geq 2n/3$, such that the subgraph of G induced by $\mathcal{I}$ is an outerplane graph.*

Chapter organization. The rest of the chapter is organized as follows. In [Section 8.1](#) we present some definitions and preliminaries. In [Section 8.2](#) we discuss more in detail the flaw in the algorithm by Borradaile, Le and Sherman-Bennett [15]. Finally, in [Section 8.3](#) we show a new proof for [Theorem 8.1](#).

8.1 Preliminaries

We refer to [Chapter 1](#) for basic definitions on graphs. Let G be a simple undirected graph. Here, we denote by $\delta_G(v)$ the degree of a vertex v in G . We recall that, for any set $\mathcal{I} \subseteq V(G)$, $G[\mathcal{I}]$ denotes the subgraph of G induced by $\mathcal{I}$. We also recall that a planar drawing is outerplanar if every vertex is incident to the outer face; an outerplanar graph is a graph that admits an outerplanar drawing; the topological information associated with such a drawing is called an outerplane embedding; and a graph together with an outerplane embedding is called an outerplane graph.

A planar drawing is *2-outerplanar* if removing all vertices incident to the outer face results in an outerplanar drawing. A *2-outerplanar graph* is a graph that admits a 2-outerplanar drawing. We call *2-outerplane embedding* the topological information associated with such a drawing, and we call *2-outerplane graph* a graph together with a 2-outerplane embedding. An induced subgraph $G[\mathcal{I}]$ of a 2-outerplane graph G inherits the embedding of G , thus we say that $G[\mathcal{I}]$ is outerplane if restricting the embedding of G to the vertices and edges of $G[\mathcal{I}]$ results in an outerplane graph. A 2-outerplane graph is *internally-triangulated* if all its internal faces are 3-cycles. Given a 2-outerplane graph G , we denote by L_1 the set of vertices of G that lie on the outer face, and by L_2 the set of vertices of G that do not lie on the outer face, i.e., $L_2 = V(G) - L_1$. In our proofs we will exploit several times the following observation. We say that two outerplane subgraphs of a 2-outerplane graph are *each in the outer face of the other one* if every vertex and edge of each graph is in the outer face or on the boundary of the outer face of the other graph.

Observation 8.2. *Let G_1 and G_2 be two outerplane subgraphs of a 2-outerplane graph such that: (i) each of G_1 and G_2 is in the outer face of the other one;*

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

and (ii) G_1 and G_2 do not share any vertex, or share a single vertex, or share a single edge. Then the subgraph of G composed of G_1 and G_2 is outerplane.

A *cutvertex* of a connected graph G is a vertex whose removal disconnects G . A connected graph is *biconnected* if it contains no cutvertex. A *biconnected component* (or *block*) of a graph G is a maximal (in terms of vertices and edges) biconnected subgraph of G . A biconnected graph or a block is *trivial* if it is a single edge, and *non-trivial* otherwise. For a connected graph G , the *block-cutvertex tree* T_G is a tree that describes the arrangement of biconnected components of G . It contains a B-node for each block of G and a C-node for each cutvertex of G . A B-node B and a C-node c are adjacent in T_G if the cutvertex corresponding to c is a vertex of the block corresponding to B . Note that the leaves of T_G are all B-nodes. We often identify a cutvertex with its corresponding C-node and a block with its corresponding B-node.

We are going to exploit the following lemma.

Lemma 8.3. *Let G be a 2-outerplane graph with $n \geq 3$ vertices. It is possible to add edges to G so that it becomes an internally-triangulated 2-outerplane non-trivial biconnected graph.*

Proof. Throughout the proof, let f_G denote the outer face of G and W_G the boundary of f_G . We add edges to G in 4 steps.

(Step 1) First, we augment G so that W_G is connected. This is done as follows. While there exist two connected components C_1 and C_2 that contain vertices v_1 and v_2 in L_1 , respectively, we add the edge v_1v_2 to G and embed it in f_G . This augmentation maintains the embedding 2-outerplane, since the edge v_1v_2 does not prevent any vertex from being incident to f_G and is incident to two vertices in L_1 . Once Step 1 is concluded, W_G is connected.

(Step 2) Second, we augment G so that W_G is a cycle. This is done as follows. We traverse W_G in clockwise direction. Whenever we encounter two consecutive edges v_1u and uv_2 that belong to distinct blocks B_1 and B_2 of G , respectively, we add the edge v_1v_2 to G and embed it in f_G , so that the path v_1uv_2 is replaced by the edge v_1v_2 in W_G . This augmentation maintains the embedding 2-outerplane, since the edge v_1v_2 does not prevent any vertex from being incident to f_G and is incident to two vertices in L_1 . In particular, an incidence of u on f_G is turned into an incidence of u on the internal face delimited by the cycle v_1uv_2 ; however, since the edges v_1u and uv_2 are incident to f_G , the edges uw_1 and uw_2 that respectively follow and precede v_1u and uv_2 in clockwise direction along the boundaries of B_1 and B_2 (such edges are

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

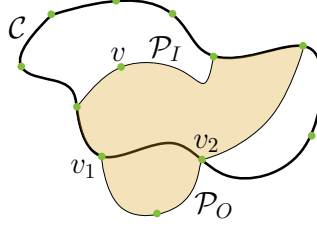


Figure 8.1: Illustration for the correctness of Step 4 in the proof of [Lemma 8.3](#). The cycle C is represented by a thick line and the face f is shaded orange.

again uw_1 and/or uw_2 if B_1 and/or B_2 are trivial), are also incident to f_G , hence u remains incident to f_G . Once Step 2 is concluded, W_G is a cycle.

(Step 3) Third, while G contains an internal face f whose boundary contains two vertices v_1 and v_2 such that at least one of them is in L_1 and such that the edge v_1v_2 does not belong to G , we add to G the edge v_1v_2 and embed it inside f . This augmentation maintains the embedding 2-outerplane, since the edge v_1v_2 does not prevent any vertex from being incident to f_G , given that it is embedded inside an internal face of G , and since it is incident to at least one vertex in L_1 . Once Step 3 is concluded, every vertex in L_1 that is on the boundary of an internal face f of G is adjacent to all the vertices on the boundary of f .

(Step 4) Finally, for every internal face f whose boundary contains at least 4 distinct vertices, we add an edge between two vertices v_1 and v_2 that are incident to f and that are not already adjacent in G (note that v_1 and v_2 are in L_2 , as otherwise Step 3 would not be concluded). The existence of such a pair of vertices is a standard argument: The vertices incident to f cannot induce a clique, since they are at least 4 and since they induce an outerplanar subgraph of G . We prove that this augmentation maintains the embedding 2-outerplane. Suppose, for a contradiction, that this is not the case; refer to [Figure 8.1](#). The edge v_1v_2 does not prevent any vertex from being incident to f_G , since it is embedded inside an internal face of G . Hence, if the embedding is not 2-outerplane, there exists a cycle C in $G[L_2]$ such that the bounded region $\mathcal{R}$ delimited by C contains a vertex v in its interior. Suppose that C is *minimal*, i.e., no cycle $C' \neq C$ is such that the bounded region delimited by C' contains v in its interior and all the edges of C' are in the closure of $\mathcal{R}$. Indeed, if such a cycle C' exists, we can use it in place of C ; the repetition of this argument eventually leads to a minimal cycle C . Note that v_1v_2 is an edge

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

of $\mathcal{C}$, as otherwise G would not be 2-outerplane even before the augmentation. Also, f is externally delimited by a cycle which consists of two paths between v_1 and v_2 , one, say $\mathcal{P}_I$, whose vertices and edges are all in the closure of $\mathcal{R}$ and one, say $\mathcal{P}_O$, whose vertices and edges are all in the closure of the complement of $\mathcal{R}$. We have that v is a vertex of $\mathcal{P}_I$, as otherwise $\mathcal{P}_I$, together with the edge v_1v_2 or with the path $\mathcal{P}$ obtained from $\mathcal{C}$ by removing the edge v_1v_2 , would create a cycle $\mathcal{C}'$ such that the bounded region delimited by $\mathcal{C}'$ contains v in its interior and all the edges of $\mathcal{C}'$ are in the closure of $\mathcal{R}$, contradicting the minimality of $\mathcal{C}$. This implies that $\mathcal{P}_O$ consists only of vertices in L_2 , as if it contained a vertex u in L_1 , then the edge uv would not belong to G , since v and u are one inside and one outside $\mathcal{C}$, hence we would be in Step 3 and the edge uv , rather than v_1v_2 , would be added to G . Then the paths $\mathcal{P}$ and $\mathcal{P}_O$ define a walk in $G[L_2]$ which bounds a region which contains v in its interior. Hence, G is not 2-outerplane before the augmentation, a contradiction.

Once Step 4 is concluded, G is internally-triangulated. Since $n \geq 3$ and W_G is a cycle, we have that G is a non-trivial biconnected graph and concludes the proof of the lemma. $\square$

8.2 On the Proof of **Theorem 8.1** by Borradaile, Le and Sherman-Bennett

In this section, we discuss the flaw in the proof of **Theorem 8.1** presented by Borradaile, Le and Sherman-Bennett [15]. There are three distinct levels for this discussion.

First, at the highest level, their general strategy, which they describe as “*To find the vertices inducing a large outerplanar graph in G , we delete vertices in L_1 until all vertices in L_2 are “exposed” to the external face*” does not, unfortunately, allow one to find a set of $2n/3$ vertices that induces an outerplane graph. Indeed, consider the 2-outerplane graph G in **Figure 8.2a**. Each vertex in L_1 prevents a distinct vertex in L_2 from being incident to the outer face. Thus, if a set of vertices in G induces an outerplane graph and contains all the vertices in L_2 , it does not contain any of the vertices in L_1 , and thus it contains at most 7 of the 11 vertices of G . Note that $7/11 < 2/3$. Furthermore, this counterexample holds even if one wants to find an induced *outerplanar*, rather than *outerplane*, graph in G . In fact, any vertex of L_1 induces, together with 4 vertices in L_2 , a subgraph of G containing $K_{2,3}$. Replicating this example as shown in **Figure 8.2b** allows us to state the following.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

Theorem 8.4. *For every positive n multiple of 11, there exists an n -vertex 2-outerplane graph G such that every set of vertices in G which induces an outerplanar graph and includes all the vertices not incident to the outer face of G contains at most $\frac{7n}{11}$ vertices.*

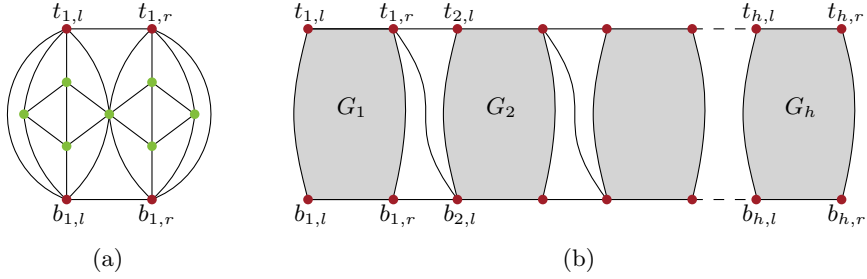


Figure 8.2: An n -vertex 2-outerplane graph on which the algorithm by Borradaile et al. fails to find a set of $2n/3$ vertices that induces an outerplanar graph. In this and the following figures, vertices depicted in red and green belong to L_1 and L_2 , respectively.

Second, at an intermediate level, the algorithm by Borradaile et al. relies on defining triples of vertices, so that each vertex in L_2 occurs in exactly one triple; each triple contains either a single vertex in L_2 and two of its neighbors in L_1 , or two vertices in L_2 and a common neighbor in L_1 . Once a vertex in L_1 is deleted, the vertices in L_2 in the same triple as that vertex are exposed on the outer face. In order to find these triples, they rely on a structural lemma ([15, Lemma 3]), which asserts that there exists a matching on the boundary of $G[L_2]$ which spans every vertex of L_2 that has a unique neighbor in L_1 . The graph G shown in Figure 8.2a is a counter-example to this statement. Indeed, L_2 consists of 7 vertices; also, 4 of such vertices have a unique neighbor in L_1 and no two of such 4 vertices are adjacent on the boundary of $G[L_2]$. Hence, no matching exists on the boundary of $G[L_2]$ which spans all such 4 vertices.

Third, at a lowest level, their proof of [15, Lemma 3] works by induction on the number of vertices with a unique neighbor in L_1 . If there is at least one of such vertices, then its neighbors on the boundary of $G[L_2]$ can be contracted into it, resulting in a graph with one less vertex with a unique neighbor in L_1 . The contraction might result in parallel edges and loops, which the authors remove from the graph. However, such a removal might result in a 2-outerplane graph which is not internally-triangulated any longer. Hence, the tools they

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

use to find the next vertices to be contracted [15, Observation 16, Lemma 1, Lemma 2] are not applicable to the obtained graph and the induction breaks down.

Remark. The described flaw in the proof of [15, Lemma 3] does not occur if $G[L_2]$ is biconnected or, more in general, a set of pairwise-disjoint biconnected components, since in this case the contractions they use preserve the property that the graph is internally-triangulated. Thus, the algorithm by Borradaile et al. successfully finds a set of $2n/3$ vertices that induces an outerplane graph if the input n -vertex graph G is such that $G[L_2]$ is a set of pairwise-disjoint biconnected components. Unfortunately, we cannot use this result as a black box in our algorithm for general 2-outerplane graphs, since our induction needs to ensure stronger properties on the large set of vertices inducing an outerplane graph.

8.3 A New Proof of Theorem 8.1

In this section, we show an algorithm that, given an n -vertex 2-outerplane graph G , computes a set $\mathcal{I} \subseteq V(G)$ with $|\mathcal{I}| \geq 2n/3$, such that $G[\mathcal{I}]$ is an outerplane graph. This proves Theorem 8.1. We will not explicitly discuss the running time of our algorithm, however, it is easy to see that it can be implemented to run in polynomial time.

The section is organized as follows. In Section 8.3.1, we introduce some definitions. In Section 8.3.2, we present the case distinction used by our inductive algorithm. Finally, in Section 8.3.3, we discuss the inductive cases of our algorithm.

8.3.1 Definitions

Our algorithm works by induction on n . In the base case, we have $n \leq 2$ and then the entire vertex set of G induces an outerplane graph. If $n \geq 3$, by Lemma 8.3 we can assume that G is an internally-triangulated non-trivial biconnected graph. Given a set of vertices $\mathcal{I} \subseteq V(G)$, we say that $\mathcal{I}$ is a *good set* if $|\mathcal{I}| \geq \frac{2}{3}n$ and if $G[\mathcal{I}]$ is outerplane.

We introduce some definitions. Let G be an internally-triangulated 2-outerplane graph with a distinguished edge e^* incident to its outer face, as the one in Figure 8.3a. Recall that the *weak dual* T of $G[L_1]$ is the tree that has a node for each internal face of $G[L_1]$ and an edge between two nodes if the corresponding faces share an edge on their boundaries; see Figure 8.3b. We

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

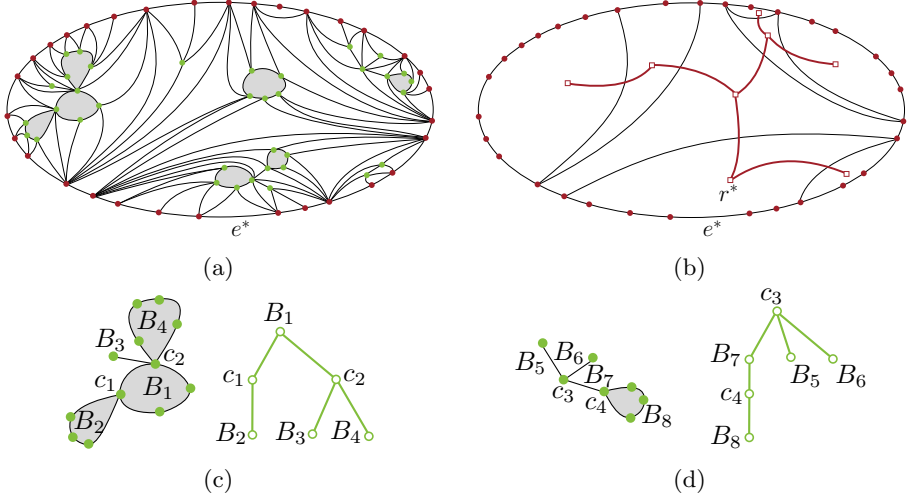


Figure 8.3: (a) An internally-triangulated 2-outerplane graph G with a distinguished edge e^* along its outer face. (b) The outerplane graph $G[L_1]$, together with its rooted weak dual tree T^* , whose nodes are empty squares and whose edges are thick and red. (c)-(d) Two terminal components, say K_c and K_d , of $G[L_2]$, together with their rooted block-cutvertex trees T_{K_c} and T_{K_d} , respectively. The extremal leaves of T_{K_c} are B_2 , B_3 , and B_4 , while B_8 is the only extremal leaf of T_{K_d} .

root T at the node r^* corresponding to the unique internal face of $G[L_1]$ that has e^* on its boundary; we denote by T^* the tree T once rooted at r^* . Consider a leaf l of T^* , which corresponds to a face f of $G[L_1]$. The face f might be also a face of G , or it might contain a connected component K of $G[L_2]$. In the latter case, we say that K is a *terminal component* of $G[L_2]$. Consider a terminal component K of $G[L_2]$ embedded in G inside a face f of $G[L_1]$ corresponding to a leaf l of T^* ; see the examples in Figures 8.3c and 8.3d. The rooting of T^* induces a rooting of the block-cutvertex tree T_K of K as follows. Let xy be the edge of $G[L_1]$ dual to the edge of T^* from l to its parent (or let $e^* = xy$ if l is the root of T^*), and let z be the vertex of K such that the 3-cycle xyz bounds an internal face of G . If z is a cutvertex of K , then T_K is rooted at the corresponding C-node, otherwise T_K is rooted at the unique B-node containing z . A leaf B in the rooted tree T_K is *extremal* if it has maximum depth (i.e., distance from the root) among the leaves of T_K . Note that the

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

children of the parent of an extremal leaf are also extremal leaves.

Consider a terminal component K of $G[L_2]$ and suppose that K is not biconnected. Let B be an extremal leaf of the rooted block-cutvertex tree T_K of K . The unique neighbor of B in T_K is called *link-vertex* of B and denoted by c_B . If B is a trivial block, it corresponds to an edge $c_B d_B$. If B is a non-trivial block, the neighbors of c_B on the boundary of the outer face of B are called *leftmost* and *rightmost* neighbors of c_B in B and denoted by u_B and v_B , where u_B , v_B , and c_B appear in this clockwise order along the outer face of B . Let u'_B be the vertex that follows u_B in clockwise order along the outer face of B and let v'_B be the vertex that follows v_B in counter-clockwise order along the outer face of B .

If B is a trivial (non-trivial) block, let $c_B \ell_B$ be the edge of G that precedes $c_B d_B$ (resp. $c_B u_B$) in clockwise direction around c_B and note that $\ell_B \in L_1$. We call ℓ_B the *left cage vertex* of B . Symmetrically, let $c_B r_B$ be the edge of G that follows $c_B d_B$ (resp. $c_B v_B$) in clockwise direction around c_B and note that $r_B \in L_1$. We call r_B the *right cage vertex* of B . The path obtained by traversing in clockwise direction the outer face of G from ℓ_B to r_B is called the *cage path* of B and denoted by $\mathcal{P}_B$. Let ℓ'_B be the vertex that precedes ℓ_B in clockwise order along the outer face of G , and let r'_B be the vertex that follows r_B in clockwise order along the outer face of G . Finally, the subgraph of G whose vertices and edges are inside or on the boundary of the cycle $\mathcal{P}_B \cup r_B c_B \ell_B$ is called the *cage graph* of B and denoted by G_B . The fact that K is a terminal component of $G[L_2]$ and that B is a leaf of T_K implies that G_B does not contain edges between two non-consecutive vertices of $\mathcal{P}_B$; hence, every internal face of G that has an edge of $\mathcal{P}_B$ on its boundary has a vertex of B as its third incident vertex.

If B is non-trivial, then we say that B is *pesky* (see Figure 8.4) if it satisfies the following properties (the first two properties actually imply the other two, but we state all four of them, so to better describe the structure of G_B):

- first, $\delta_{G_B}(\ell_B) = \delta_{G_B}(r_B) = 4$;
- second, for each internal vertex w of $\mathcal{P}_B$, we have $\delta_{G_B}(w) = 5$;
- third, u_B and v_B have only ℓ_B and r_B , respectively, as neighbors in L_1 ; and
- finally, traversing the boundary of the outer face of B in clockwise direction from u_B to v_B , the encountered vertices have alternately 1 and 2 neighbors in L_1 .

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

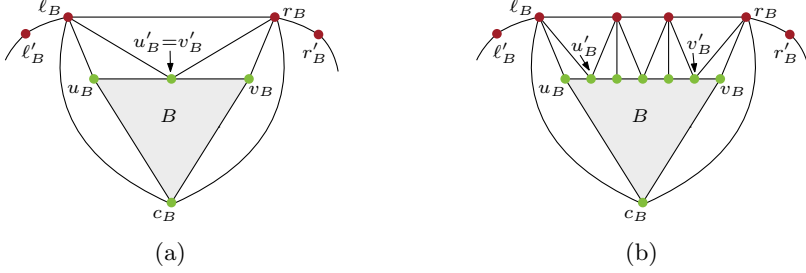


Figure 8.4: Two pesky blocks. In this and the following figures, vertices depicted in red and green belong to L_1 and L_2 , respectively.

We observe the following.

Observation 8.5. *If B is pesky, the number of vertices of B is exactly twice the number of vertices of $\mathcal{P}_B$.*

If B is non-trivial and not pesky, then we say that it is *cushy*.

8.3.2 Case distinction

We now discuss the case distinction employed by our inductive algorithm. Suppose that G is a non-trivial biconnected graph. In all the inductive cases, the basic strategy is the one of removing some vertices of the input 2-outerplane graph G , obtaining a smaller 2-outerplane graph H , of then applying induction on H , so to get a good set $\mathcal{I}_H$ for H , and of then defining a good set $\mathcal{I}$ for G by adding to $\mathcal{I}_H$ some of the initially removed vertices. Two initial inductive cases are used to deal with the situation in which G contains a vertex $v \in L_1$ whose degree in G is 2 (**Case 1**) or 3 (**Case 2**). This allows us to assume that every vertex of L_1 has degree at least 4 in G .

Next, consider a leaf l of the rooted weak dual tree T^* of $G[L_1]$, and let f be the face of $G[L_1]$ corresponding to l . Since no vertex of L_1 has degree 2 in G , it follows that f contains a terminal component K . Let xy be the edge of $G[L_1]$ dual to the edge of T^* from the node corresponding to f to its parent (or let xy be the edge e^* if the node corresponding to f is the root of T^*). Note that K is not a single vertex, otherwise all its neighbors in L_1 different from x and y (there exists at least one such a neighbor) would have degree 3 in G , which is excluded. If K is a single edge uv (**Case 3**), then f is delimited by a 3-cycle xyz such that z is adjacent to both u and v , or by a 4-cycle $xytz$, such

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

that each of t and z is adjacent to both u and v . Indeed, if f was delimited by a cycle with more than 4 vertices, at least one of them would have degree 3 in G ; likewise, if one of the vertices on the boundary of f was not adjacent to u or v , then it would have degree 3 in G .

Having ruled out that K is a vertex or an edge, it might be that K is a non-trivial biconnected graph (**Case 4**). Unfortunately, we cannot handle this case via the algorithm by Borradaile et al. [15]. Indeed, the natural choice for the graph H on which induction should be applied is the following: Remove K and the vertices in the cycle $\mathcal{C}_f$ delimiting f , except for x and y , from G , and let the resulting graph be H . However, with this approach, the induction would tell us whether x and y belong to the set of vertices inducing an outerplane graph, and the algorithm by Borradaile et al. does not guarantee this freedom of choice when applied to the 2-outerplane graph induced by the vertices inside or on the boundary of $\mathcal{C}_f$. Hence, we need to come up with a new algorithm providing guarantees on x and y .

In the remaining cases, K is not biconnected. Then recall that T_K denotes the rooted block-cutvertex tree of K . Let B be an extremal leaf of K . Suppose first that B is non-trivial. If B is cushy (**Case 5**), then we can apply a natural strategy for defining H : We remove from G all the cage graph G_B of B , except for the link-vertex c_B and, possibly, for the left and/or right cage vertices ℓ_B and r_B . If, on the other hand, B is pesky, then this simple strategy does not work (which justifies the attribute *pesky*) and we need to explore the structure of G in further depth. In particular, the next cases handle the situations in which: The edge $c_B\ell'_B$ exists and the 3-cycle $c_B\ell_B\ell'_B$ bounds an internal face of G (**Case 6**); the edge $c_Br'_B$ exists and the 3-cycle $c_Br_Br'_B$ bounds an internal face of G (**Case 6'**); the edge $c_B\ell'_B$ exists and the 3-cycle $c_B\ell_B\ell'_B$ contains in its interior a unique vertex (**Case 7**); and the edge $c_Br'_B$ exists and the 3-cycle $c_Br_Br'_B$ contains in its interior a unique vertex (**Case 7'**). Having ruled out that there is a face (Cases 6 and 6') or a K_4 (Cases 7 and 7') next to G_B , we next handle the situation in which there is a different pesky block B^* next to G_B . Formally, this happens if there exists a pesky non-trivial extremal leaf $B^* \neq B$ in K such that the link-vertex c_{B^*} of B^* is c_B and the right cage vertex r_{B^*} of B^* is ℓ_B (**Case 8**) or if there exists a pesky non-trivial extremal leaf $B^* \neq B$ in K such that the link-vertex c_{B^*} of B^* is c_B and the left cage vertex ℓ_{B^*} of B^* is r_B (**Case 8'**).

The next cases assume that B is trivial. Indeed, similarly to the case in which B is non-trivial, we can successfully apply induction if: The edge $c_B\ell'_B$ exists and the 3-cycle $c_B\ell_B\ell'_B$ bounds an internal face of G (**Case 9**); the edge $c_Br'_B$ exists and the 3-cycle $c_Br_Br'_B$ bounds an internal face of G (**Case 9'**);

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

the edge $c_B \ell'_B$ exists and the 3-cycle $c_B \ell_B \ell'_B$ contains in its interior a unique vertex (**Case 10**); and the edge $c_B r'_B$ exists and the 3-cycle $c_B r_B r'_B$ contains in its interior a unique vertex (**Case 10'**).

The described case distinction greatly simplifies the structure of G that can be assumed in the proximity of an extremal leaf B of the rooted block-cutvertex tree T_K of a terminal component K of $G[L_2]$, as stated in the following.

Lemma 8.6. *If neither of Cases 5, 6, 6', 7, 7', 8, 8', 9, 9' 10, and 10' applies, then B is the only child of the cutvertex c_B in T_K .*

Proof. Suppose, for a contradiction, that c_B has a child $B' \neq B$ in T_K . Since B is an extremal leaf of T_K , then so is B' . The edges incident to c_B that belong to the cage graph G_B of B appear consecutively in the circular order around c_B , between the edges $c_B \ell_B$ and $c_B r_B$, and likewise for the edges incident to c_B that belong to the cage graph $G_{B'}$ of B' . We define a linear order σ of the edges that are incident to c_B and that belong to blocks of K that are children of c_B in T_K , as follows. Start from any edge incident to c_B in the parent block of c_B in T_K (or start from the edge $c_B x$ if c_B is the root of T_K). Then visit in clockwise order the edges incident to c_B in G and append to σ those that belong to blocks of K that are children of c_B in T_K . We can assume that, in σ , the edges that belong to B come right after the edges that belong to B' , as otherwise a different choice for B and B' can be made, so to guarantee this property. If B or B' is non-trivial and cushy, then Case 5 applies, a contradiction. Otherwise, consider the face incident to the edge $c_B \ell_B$ and outside the cage graph G_B ; let z be the third vertex incident to the face. If $z \in L_1$, then $z = \ell'_B$, as otherwise K would not be a terminal component of $G[L_2]$, hence we would be in Case 6 or Case 9, depending on whether B is non-trivial or trivial, respectively, a contradiction. Hence, we have $z \in L_2$, which implies that $z \in V(B')$. If B' is trivial, then Case 7 or Case 10 applies, depending on whether B is non-trivial or trivial, respectively, a contradiction. If B' is non-trivial and pesky, and B is trivial, then Case 7' applies, a contradiction. Finally, if both B and B' are non-trivial and pesky, then Case 8 applies, a contradiction. Since in every case we get a contradiction, the lemma follows. $\square$

By **Lemma 8.6** and since K is not biconnected, c_B has a parent B-node B_P in T_K . We can handle directly the case in which B_P is a trivial block, both if B is non-trivial (**Case 11**) and if it is trivial (**Case 12**). In order to discuss Case 11 and Case 12, it will be useful to exploit the following property.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

Lemma 8.7. *Suppose that B_P is a trivial block, corresponding to an edge $c_B p$. Then the edges $p\ell_B$ and pr_B belong to G , and the cycles $p\ell_B c_B$ and $pr_B c_B$ delimit faces of G .*

Proof. Consider the face incident to the edge $c_B \ell_B$ and outside the cage graph G_B ; let z be the third vertex incident to the face. We prove that $z \in L_2$. Suppose, for a contradiction, that $z \in L_1$. If $z = \ell'_B$, then Case 6 or Case 9 applies, depending on whether B is non-trivial or trivial, respectively, a contradiction. If $z \neq \ell'_B$, then $\ell_b \ell'_B$ is an internal edge of $G[L_1]$, hence it is dual to an edge e of T^* ; one of the end-vertices of e is the node l of T^* corresponding to the face f of $G[L_1]$ containing K . If l is the parent of the other end-vertex of e in T^* , then K is not a terminal component of $G[L_2]$, a contradiction. If l is a child of the other end-vertex of e in T^* , then $\ell_b \ell'_B$ is the edge xy . However, this implies that T_K is rooted at c_B , a contradiction to the fact that c_B has a parent B_P in T_K . This proves that $z \in L_2$. It follows that $z = p$, as otherwise the edge $c_B z$ would be an edge between two vertices of different connected components of $G[L_2]$ or an edge between two different blocks of K , which is a contradiction in both cases. It can be symmetrically proved that the edge pr_B belongs to G and the cycle $pr_B c_B$ delimits an internal face of G . $\square$

We are now ready to discuss the final inductive case (**Case 13**), which occurs when neither of Cases 1-12 applies in the entire graph G (in particular, Cases 5-12 do not apply to any extremal leaf in any terminal component of $G[L_2]$). The next lemma describes the structure that we are guaranteed to encounter in Case 13.

Lemma 8.8. *If neither of Cases 1-12 applies, then there exists a terminal component K of $G[L_2]$ that contains a biconnected component B_P that satisfies the following properties.*

- *First, B_P is non-trivial and is not a leaf of T_K .*
- *Second, each child c_B of B_P in T_K has a unique child B , which is an extremal leaf of T_K and is either a trivial block or a non-trivial pesky block.*
- *Third, for each child B of a child c_B of B_P , let c_B^l and c_B^r be the vertices that precede and follow c_B in clockwise order along the boundary of B_P ; then the edges $c_B^l \ell_B$ and $c_B^r r_B$ belong to G , and the cycles $c_B^l \ell_B c_B$ and $c_B^r r_B c_B$ delimit internal faces of G .*

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

Proof. Since Case 1 does not apply, we have that $G[L_2]$ contains a terminal component. Let then K be any terminal component of $G[L_2]$. Since Cases 2, 3, and 4 do not apply to K , it follows that K is not biconnected. Let then B be an extremal leaf of T_K . Since Case 5 does not apply, B is either a trivial block or a non-trivial pesky block. Since Cases 5, 6, 6', 7, 7', 8, 8', 9, 9', 10, and 10' do not apply, by [Lemma 8.6](#), we have that B is the only child of its parent c_B in T_K . We can now define B_P as the parent of c_B in T_K . By construction, B_P is not a leaf of T_K . Also, since Cases 11 and 12 do not apply, B_P is non-trivial. Since B is an extremal leaf, every child of a child of B_P is also an extremal leaf, and then by the same arguments as for B , it is either a trivial block or a non-trivial pesky block and it is the only child of its parent in T_K . It remains to prove the third item. This proof is very similar to the proof of [Lemma 8.7](#). Consider any child B of a child c_B of B_P , and let c_B^l be the vertex that precedes c_B in clockwise order along the boundary of B_P . Consider the face incident to the edge $c_B c_B^l$ and outside the cage graph G_B ; let z be the third vertex incident to the face. We prove that $z \in L_2$. Suppose, for a contradiction, that $z \in L_1$. If $z = \ell'_B$, then Case 6 or Case 9 applies, depending on whether B is non-trivial or trivial, respectively, a contradiction. If $z \neq \ell'_B$, then $\ell_b \ell'_B$ is an internal edge of $G[L_1]$, hence it is dual to an edge e of T^* ; one of the end-vertices of e is the node l of T^* corresponding to the face f of $G[L_1]$ containing K . If l is the parent of the other end-vertex of e in T^* , then K is not a terminal component of $G[L_2]$, a contradiction. If l is a child of the other end-vertex of e in T^* , then $\ell_b \ell'_B$ is the edge xy . However, this implies that T_K is rooted at c_B , a contradiction to the fact that c_B has a parent B_P in T_K . This proves that $z \in L_2$. It follows that $z = c_B^l$, as otherwise the edge $c_B z$ would be an edge between two vertices of different connected components of $G[L_2]$, or an edge between two different blocks of K , or an edge between two vertices of the same block B of K not belonging to B , which is a contradiction in every case. It can be symmetrically proved that the edge $c_B^r r_B$ belongs to G and the cycle $c_B^r r_B c_B$ delimits an internal face of G , where c_B^r is the vertex that follows c_B in clockwise order along the boundary of B_P . $\square$

Finally, it remains to observe that one of the inductive cases always applies; indeed, if neither of Cases 1–12 applies, then Case 13 does. Note that the graph H constructed in order to apply induction might be not biconnected, however by [Lemma 8.3](#) it can be augmented to a non-trivial biconnected graph by only adding edges to it, as long as it has at least three vertices (while it encounters the base case if it has at most two vertices).

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

8.3.3 Inductive cases

We now discuss the inductive cases. Each inductive case assumes that neither of the previous cases applies.

Case 1: There exists a vertex $v \in L_1$ such that $\delta_G(v) = 2$. Let u and z be the neighbors of v in G , and observe that they belong to L_1 (see [Figure 8.5a](#)). We construct an $(n - 1)$ -vertex 2-outerplane graph H from G by removing v and its incident edges. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \{v\}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + 1 \geq 2(n - 1)/3 + 1 > 2n/3$. Also, $G[\mathcal{I}]$ is outerplane by [Observation 8.2](#), since $G[\mathcal{I}_H]$ and $G[v \cup (\{u, z\} \cap \mathcal{I}_H)]$ are outerplane, are each in the outer face of the other one, and share either no vertex, or a single vertex, or a single edge.

Case 2: There exists a vertex $v \in L_1$ such that $\delta_G(v) = 3$. Let u and z be the neighbors of v along the outer face of G ; hence, u and z belong to L_1 . The third neighbor, v' , of v might belong to L_1 or L_2 . We further distinguish three cases.

Case 2.1: The edge uz does not belong to G (see [Figure 8.5b](#)). We construct an $(n - 1)$ -vertex 2-outerplane graph H from G by removing v and its incident edges, and by adding the edge uz in the outer face of H in place of the path uvz . We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \{v\}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + 1 \geq 2(n - 1)/3 + 1 > 2n/3$. Suppose, for a contradiction, that $G[\mathcal{I}]$ contains a cycle $\mathcal{C}$ which contains a vertex $w \in \mathcal{I}$ in its interior. Note that $w \neq v$, since v is incident to the outer face of G , hence $w \in \mathcal{I}_H$. Also, v is one of the vertices of $\mathcal{C}$, as otherwise $G[\mathcal{I}_H]$ would also contain $\mathcal{C}$, and $\mathcal{I}_H$ would not be a good set. The neighbors of v in $\mathcal{C}$ are two vertices among u , z , and v' . Note that H contains all three edges uz , uv' , and $v'z$, the first one by construction and the other two since G is internally-triangulated. Hence, the length-2 path in $\mathcal{C}$ that has v as middle vertex can be replaced by the edge between its end-vertices, resulting in a cycle which also contains w in its interior and whose vertices all belong to $\mathcal{I}_H$, which implies that $\mathcal{I}_H$ is not a good set, a contradiction.

Case 2.2: The edge uz belongs to G and the subgraph G_{uvz} of G induced by the vertices inside or on the boundary of the 3-cycle uvz has at least 6 vertices (see [Figure 8.5c](#)). Note that the assumption that uz belongs to G implies that $v' \in L_2$. Let H be the subgraph of G induced by $V(G) - V(G_{uvz})$. We

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

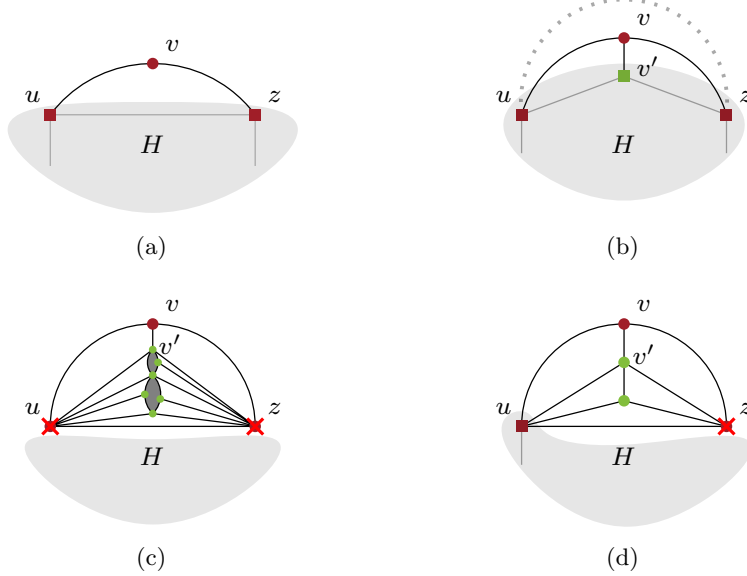


Figure 8.5: Illustrations for (a) Case 1, (b) Case 2.1, (c) Case 2.2, and (d) Case 2.3. In this and the following figures, squared vertices belong to the graph H (shown with gray edges and a shaded area which hides most of its vertices and edges) on which induction is applied, while rounded vertices do not belong to H . A cross on a vertex not belonging to H indicates that it is not selected to be in the set $\mathcal{I}$ inducing an outerplane graph. In (b), the dotted curve represents an edge added to H for the induction and the vertex v' might belong to L_1 or L_2 .

apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I}$ as $\mathcal{I}_H$ plus all the vertices of G_{uvz} different from u and z . We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(G_{uvz})| - 2$. Note that $|V(H)| = n - |V(G_{uvz})|$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - |V(G_{uvz})|)/3$. It follows that $|\mathcal{I}| \geq 2(n - |V(G_{uvz})|)/3 + |V(G_{uvz})| - 2 = 2n/3 + |V(G_{uvz})|/3 - 2 \geq 2n/3$, where the last inequality comes from the fact that $|V(G_{uvz})| \geq 6$. By [Observation 8.5](#), we have that $G[\mathcal{I}]$ is an outerplane graph, since it is composed of the outerplane graphs $G[\mathcal{I}_H]$ and $G_{uvz} - \{u, z\}$, which are each in the outer face of the other one and do not share any vertex. Note that $G_{uvz} - \{u, z\}$ is indeed an outerplane graph since it is composed of the outerplane graph $G[V(G_{uvz}) \cap L_2]$ augmented

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

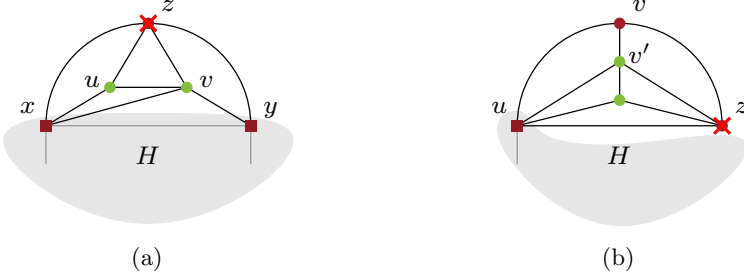


Figure 8.6: Illustrations for (a) Case 3.1 and (b) Case 3.2.

with the edge vv' in its outer face.

Case 2.3: The edge uz belongs to G and G_{uvz} has at most 5 vertices (see Figure 8.5d). Let H be the subgraph of G induced by $\{u\} \cup (V(G) - V(G_{uvz}))$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I}$ as $\mathcal{I}_H$ plus all the vertices of G_{uvz} different from u and z . Note that u is in $\mathcal{I}$ if and only if it belongs to $\mathcal{I}_H$, while z is not in $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(G_{uvz})| - 2$. Note that $|V(H)| = n - |V(G_{uvz})| + 1$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - |V(G_{uvz})| + 1)/3$. It follows that $|\mathcal{I}| \geq 2(n - |V(G_{uvz})| + 1)/3 + |V(G_{uvz})| - 2 = 2n/3 + |V(G_{uvz})|/3 - 4/3 \geq 2n/3$, where the last inequality comes from the fact that $|V(G_{uvz})| \geq 4$, given that v has degree 3. Note that $G[\mathcal{I}_H]$ is outerplane by induction and that the subgraph of G_{uvz} induced by the vertices in $\mathcal{I}$ is either a path in the outer face of $G[\mathcal{I}_H]$, if $u \notin \mathcal{I}$, or a biconnected outerplane graph whose vertex set intersects with the one of $G[\mathcal{I}_H]$ only at u , if $u \in \mathcal{I}$. Hence, by Observation 8.2, we have that $G[\mathcal{I}]$ is an outerplane graph.

In the remaining cases, l denotes a leaf of the rooted weak dual tree T^* of $G[L_1]$, where l corresponds to a face f of $G[L_1]$ and contains a terminal component K of $G[L_2]$. Let xy be the edge of $G[L_1]$ dual to the edge of T^* from the node corresponding to f to its parent (or let xy be the edge e^* if the node corresponding to f is the root of T^*).

Case 3: K is a single edge uv . Recall that f is delimited by a 3-cycle xyz such that z is adjacent to both u and v , or by a 4-cycle $xytz$, such that each of t and z is adjacent to both u and v . We distinguish these cases.

Case 3.1: The face f of $G[L_1]$ is delimited by a 3-cycle xyz such that z is adjacent to both u and v (see Figure 8.6a). Let H be the subgraph of G induced by $V(G) - \{u, v, z\}$. We apply induction on H , so to find a good

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \{u, v\}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + 2$. Note that $|V(H)| = n - 3$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - 3)/3$. It follows that $|\mathcal{I}| \geq 2(n - 3)/3 + 2 = 2n/3$. Also, $G[\mathcal{I}_H]$ is outerplane by induction and the subgraph of G induced by u, v , and by $\{x, y\} \cap \mathcal{I}_H$ is outerplane, as well, given that z is a neighbor of all of u, v, x, y . Since $G[\mathcal{I}_H]$ and $G[\{u, v\} \cup (\{x, y\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one, and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 3.2: The face f of $G[L_1]$ is delimited by a 4-cycle $xytz$, such that each of t and z is adjacent to both u and v (see [Figure 8.6b](#)). Let H be the subgraph of G induced by $V(G) - \{u, v, t, z\}$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \{u, v, t\}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + 3$. Note that $|V(H)| = n - 4$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - 4)/3$. It follows that $|\mathcal{I}| \geq 2(n - 4)/3 + 3 > 2n/3$. Also, $G[\mathcal{I}_H]$ is outerplane by induction; also, the subgraph of G induced by u, v, t , and by $\{x, y\} \cap \mathcal{I}_H$ is outerplane, as well, given that z is adjacent to all of u, v, t , and x , and given that y belong to L_1 . Since $G[\mathcal{I}_H]$ and $G[\{u, v, t\} \cup (\{x, y\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one, and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

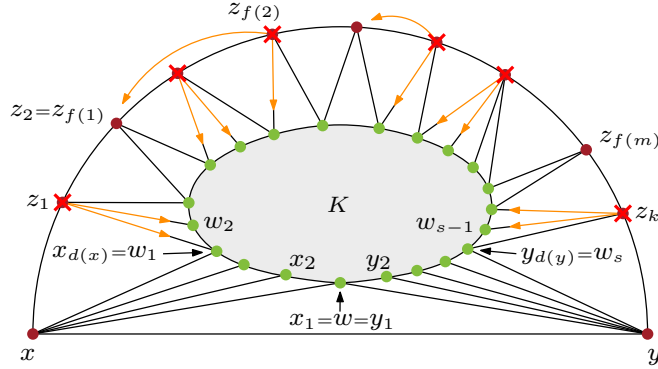


Figure 8.7: Illustration for Case 4. The result of applying the basic algorithm to L to compute $\mathcal{I}_L$. In this and the following figures, orange arrows indicate the charging of vertices that are not in $\mathcal{I}_L$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

Case 4: K is a non-trivial biconnected graph. Recall that $\mathcal{C}_f$ denotes the cycle delimiting the boundary of the face f of $G[L_1]$ containing K . Since K is a terminal component, all the edges of $\mathcal{C}_f$, except possibly for xy , are incident to the outer face of G . Denote by L the graph $G[V(\mathcal{C}_f) \cup V(K)]$ and let $n_L = |V(L)|$. Before describing how to apply induction, we need to prove the following lemma.

Lemma 8.9. *There exists a set $\mathcal{I}_L \subset V(L)$ such that $\{x, y\} \cap \mathcal{I}_L = \emptyset$ and such that (at least) one of the following properties is satisfied:*

- Property A: $|\mathcal{I}_L| \geq 2n_L/3$ and $G[\mathcal{I}_L]$ is an outerplane graph;
- Property B: $|\mathcal{I}_L| \geq 2(n_L - 1)/3$ and $G[\mathcal{I}_L \cup \{x\}]$ is an outerplane graph;
- Property C: $|\mathcal{I}_L| \geq 2(n_L - 1)/3$ and $G[\mathcal{I}_L \cup \{y\}]$ is an outerplane graph;
and
- Property D: $|\mathcal{I}_L| \geq 2(n_L - 2)/3$ and $G[\mathcal{I}_L \cup \{x, y\}]$ is an outerplane graph.

Proof. Refer to [Figure 8.7](#). Suppose, w.l.o.g., that x comes right after y in clockwise order along the boundary of the outer face of L . Then let $y, x, z_1, \dots, z_k$ be the clockwise order of the vertices along the outer face of L . The *internal degree* $d(z)$ of a vertex z on the boundary of L is the number of its neighbors in K . Note that, since Cases 1 and 2 do not apply, the internal degree of every vertex among $z_1, \dots, z_k$ is at least two. Let also $z_{f(1)}, z_{f(2)}, \dots, z_{f(m)}$ be the sequence $z_1, \dots, z_k$ restricted to the $m \geq 0$ vertices with internal degree 2. Let xyw be the internal face of L incident to the edge xy , and let $x_1 = y_1 = w, x_2, \dots, x_{d(x)} = w_1, w_2, \dots, w_s = y_{d(y)}, y_{d(y)-1}, \dots, y_2$ be the clockwise order of the vertices along the outer face of K , where $x_1, x_2, \dots, x_{d(x)}$ are the neighbors of x and $y_{d(y)}, y_{d(y)-1}, \dots, y_1$ are the neighbors of y .

We start by defining a *basic algorithm* for the construction of the required set $\mathcal{I}_L$. The algorithm is as follows. First, all the vertices of K are in $\mathcal{I}_L$. Second, x and y are not in $\mathcal{I}_L$. Third, each vertex z_j with $d(z_j) \geq 3$ is not in $\mathcal{I}_L$. Finally, a vertex $z_{f(j)}$ with internal degree 2 is in $\mathcal{I}_L$ if j is odd and not in $\mathcal{I}_L$ if j is even. Note that $\{x, y\} \cap \mathcal{I}_L = \emptyset$, as required.

Our actual algorithm for the definition of $\mathcal{I}_L$ starts from the set constructed by the basic algorithm and then it might remove $z_{f(m)}$ from it. We distinguish several cases, based on the values of $d(x)$ and $d(y)$, however it is convenient to first factor out some parts of the proof that are common to the distinct cases.

First, in order to prove that $\mathcal{I}_L$ satisfies Property A, B, C, or D, we need to show that $G[\mathcal{I}_L]$, or $G[\mathcal{I}_L \cup \{x\}]$, or $G[\mathcal{I}_L \cup \{y\}]$, or $G[\mathcal{I}_L \cup \{x, y\}]$ is outerplane, respectively. For that, it suffices to show that, for each vertex in K , there exists

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

a neighbor among $y, x, z_1, \dots, z_k$ which is not in $\mathcal{I}_L$, or not in $\mathcal{I}_L \cup \{x\}$, or not in $\mathcal{I}_L \cup \{y\}$, or not in $\mathcal{I}_L \cup \{x, y\}$, respectively. This proof strategy is reminiscent of [15]. Consider a vertex w_i , for some $2 \leq i \leq s-1$. If w_i is a neighbor of a vertex z_j with $d(z_j) \geq 3$, then by the construction of the basic algorithm z_j is not in $\mathcal{I}_L$, hence it is the desired neighbor of w_i . Each remaining vertex w_i with $2 \leq i \leq s-1$ is only neighbor of vertices z_j with $d(z_j) = 2$. Since L is internally-triangulated, it follows that w_i is a neighbor of two vertices $z_{f(j)}$ and $z_{f(j+1)}$ with $d(z_{f(j)}) = d(z_{f(j+1)}) = 2$ that are consecutive along the boundary of L . By the construction of the basic algorithm, one of them is not in $\mathcal{I}_L$, hence it is the desired neighbor of w_i . The proof that the vertices x_i and y_i have a neighbor among $y, x, z_1, \dots, z_k$ which is not in $\mathcal{I}_L$, or not in $\mathcal{I}_L \cup \{x\}$, or not in $\mathcal{I}_L \cup \{y\}$, or not in $\mathcal{I}_L \cup \{x, y\}$ will be slightly different on a case-by-case basis.

Second, in order to prove that $\mathcal{I}_L$ satisfies Property A, B, C, or D, we need to show that $|\mathcal{I}_L| \geq 2n_L/3$, or $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, or $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, or $|\mathcal{I}_L| \geq 2(n_L - 2)/3$, respectively. This is done by *charging* each vertex of L not in $\mathcal{I}_L$, or each vertex of L not in $\mathcal{I}_L$ and different from x , or each vertex of L not in $\mathcal{I}_L$ and different from y , or each vertex of L not in $\mathcal{I}_L$ and different from x and y , respectively, to two vertices in $\mathcal{I}_L$, so that each vertex in $\mathcal{I}_L$ is charged with at most one vertex. We charge each vertex z_j with $d(z_j) \geq 3$ to its first two neighbors in the sequence $w_1, w_2, \dots, w_s$. We charge each vertex $z_{f(j)}$ with $d(z_{f(j)}) = 2$ and with j even to $z_{f(j-1)}$ and to its first neighbor in the sequence $w_1, w_2, \dots, w_s$; note that $z_{f(j-1)} \in \mathcal{I}_L$. The charge of x and y will be slightly different on a case-by-case basis; also, whenever the algorithm removes $z_{f(m)}$ from $\mathcal{I}_L$, we need to charge that vertex to two vertices in $\mathcal{I}_L$, as well. Note that each vertex $z_{f(j)} \in \mathcal{I}_L$ is charged with at most one vertex, namely $z_{f(j+1)}$, if such a vertex exists. Also, if a vertex w_i is charged as the second neighbor in the sequence $w_1, w_2, \dots, w_s$ of a vertex z_j with $d(z_j) \geq 3$, then it is not charged with any other vertex, since z_j is its only neighbor in $z_1, \dots, z_k$. Furthermore, if a vertex w_i is charged as the first neighbor in the sequence $w_1, w_2, \dots, w_s$ of a vertex z_j , then it is not charged as the first neighbor in the sequence $w_1, w_2, \dots, w_s$ of a different vertex $z_{j'}$. Indeed, if w_i were charged as the first neighbor in the sequence $w_1, w_2, \dots, w_s$ of two distinct vertices z_j and $z_{j'}$ with $j < j'$, it would be the only neighbor of z_j in K , hence the degree of z_j in G would be 3, contradicting the fact that Case 2 does not apply. Hence, each vertex in the sequence $w_1, w_2, \dots, w_s$ is charged with at most one vertex. Finally, observe that the vertices $x_{d(x)-1}, x_{d(x)-2}, \dots, x_1 = y_1, y_2, \dots, w_s = y_{d(y)}$ have not yet been charged with any vertex, while $w_1 = x_{d(x)}$ might have been charged

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

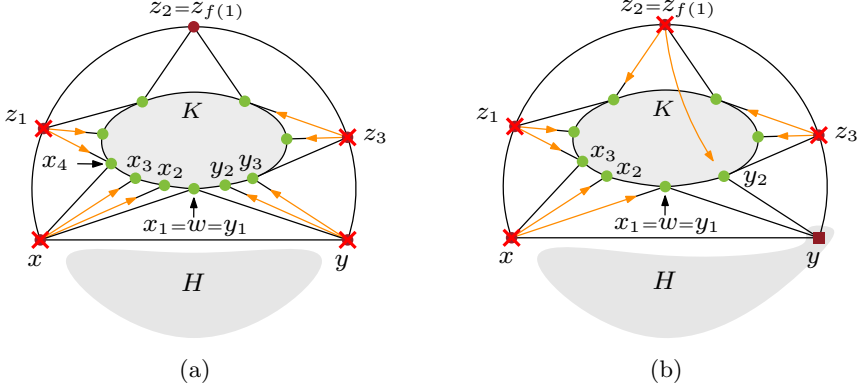


Figure 8.8: Illustrations for Case 4 (a) if $d(x) + d(y) \geq 6$ and (b) if $d(x) + d(y) = 5$.

with z_1 , if $d(z_1) \geq 3$; also, for each j odd with $1 \leq j \leq m$, the first neighbor in the sequence $w_1, w_2, \dots, w_s$ of $z_{f(j)}$ has not yet been charged with any vertex. We now present our case distinction.

- Suppose first that $d(x) + d(y) \geq 6$ (see Figure 8.8a). This implies that x and y have a total of at least 5 distinct neighbors in K . Indeed, w contributes to both $d(x)$ and $d(y)$; also, x and y do not have any common neighbor w' other than w , as otherwise, by planarity, K would lie inside the cycle $xwyw'$, thus $z_1, \dots, z_k$ would not have any neighbor in K other than w' , and hence they would have degree 3 in G , contradicting the fact that Case 2 does not apply. We define a set $\mathcal{I}_L$ by means of the basic algorithm and prove that it satisfies Property A. In order to prove that $G[\mathcal{I}_L]$ is outerplane, it suffices to observe that the neighbor x of the vertices x_i is not in $\mathcal{I}_L$ and the neighbor y of the vertices y_i is not in $\mathcal{I}_L$. In order to prove that $|\mathcal{I}_L| \geq 2n_L/3$, we charge x and y to the first and to the last two vertices in the sequence $x_{d(x)-1}, x_{d(x)-2}, \dots, x_1 = y_1, y_2, \dots, y_{d(y)}$, respectively. Since such sequence contains at least 4 vertices, each of $x_{d(x)-1}, x_{d(x)-2}, \dots, x_1 = y_1, y_2, \dots, y_{d(y)}$ is charged with at most one of x and y .
- Suppose next that $d(x) + d(y) = 5$ (see Figure 8.8b). Assume that $d(x) > d(y)$, as the case $d(x) < d(y)$ can be handled symmetrically by mirroring the embedding of G . We define a set $\mathcal{I}_L$ by means of the basic algorithm, however if m is odd, we remove $z_{f(m)}$ from $\mathcal{I}_L$. We prove that $\mathcal{I}_L$ satisfies

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

Property C (if $d(x) < d(y)$, rather than $d(x) > d(y)$, we can define a set $\mathcal{I}_L$ that satisfies Property B). In order to prove that $G[\mathcal{I}_L \cup \{y\}]$ is outerplane, we observe that the neighbor x of the vertices x_i is not in $\mathcal{I}_L \cup \{y\}$. For the vertices y_i , we cannot rely on y , which is in $\mathcal{I}_L \cup \{y\}$. However, y_1 is also a neighbor of x , which is not in $\mathcal{I}_L \cup \{y\}$. Furthermore, y_2 , if it exists, is also a neighbor of z_k , which is not in $\mathcal{I}_L \cup \{y\}$; this is obvious if $d(z_k) \geq 3$ and it comes from the choice of excluding from $\mathcal{I}_L$ the vertex $z_{f(m)}$ if $d(z_k) = 2$. In order to prove that $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, we charge x to the 2 vertices in the sequence $x_{d(x)-1}, x_{d(x)-2}, \dots, x_1 = y_1, y_2, \dots, y_{d(y)-1}$ (more precisely, this sequence is x_3, x_2 if $d(x) = 4$ and $d(y) = 1$, and it is $x_2, x_1 = y_1$ if $d(x) = 3$ and $d(y) = 2$). Also, we charge $z_{f(m)}$ to its first neighbor in the sequence $w_1, w_2, \dots, w_s$ and to $y_{d(y)}$. Note that both such vertices were previously uncharged.

- *Suppose next that $d(x) + d(y) = 4$. Assume that $d(x) \geq d(y)$, as the case $d(x) < d(y)$ can be handled symmetrically by mirroring the embedding of G . We need to further brake down this case into three subcases.*
 - *First, suppose that $d(x) = 3$ (see Figure 8.9a). We define a set $\mathcal{I}_L$ by means of the basic algorithm and prove that it satisfies Property C (if $d(y) = 3$, rather than $d(x) = 3$, we can define a set $\mathcal{I}_L$ that satisfies Property B). In order to prove that $G[\mathcal{I}_L \cup \{y\}]$ is outerplane, we observe that the neighbor x of the vertices $x_1 = y_1, x_2, x_3$ is not in $\mathcal{I}_L \cup \{y\}$. In order to prove that $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, we charge x to x_1 and x_2 . Note that both such vertices were previously uncharged.*
 - *Second, suppose that $d(x) = 2$ and m is even (see Figure 8.9b). We define a set $\mathcal{I}_L$ by means of the basic algorithm and prove that it satisfies Property C (we can similarly define a set $\mathcal{I}_L$ that satisfies Property B). In order to prove that $G[\mathcal{I}_L \cup \{y\}]$ is outerplane, we observe that the neighbor x of the vertices $x_1 = y_1, x_2$ is not in $\mathcal{I}_L \cup \{y\}$. Also, the neighbor z_k of y_2 is not in $\mathcal{I}_L \cup \{y\}$; this is true either because $d(z_k) \geq 3$, or because z_k coincides with $z_{f(m)}$ and m is even, by assumption. In order to prove that $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, we charge x to y_1 and y_2 . Note that both such vertices were previously uncharged.*
 - *Third, suppose that $d(x) = 2$ and m is odd (see Figure 8.9c). We define a set $\mathcal{I}_L$ by means of the basic algorithm and prove that it satisfies Property A. In order to prove that $G[\mathcal{I}_L]$ is outerplane, we observe that the neighbor x of the vertices x_1, x_2 is not in $\mathcal{I}_L$ and*

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

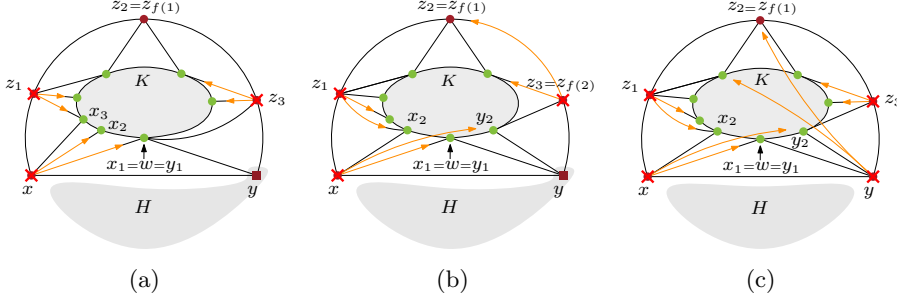


Figure 8.9: Illustrations for Case 4 when $d(x) + d(y) = 4$ (a) if $d(x) = 3$, (b) if $d(x) = 2$ and m is even, and (c) if $d(x) = 2$ and m is odd.

the neighbor y of the vertices y_1, y_2 is not in $\mathcal{I}_L$. In order to prove that $|\mathcal{I}_L| \geq 2n_L/3$, we charge x to y_1 and y_2 and y to $z_{f(m)}$ and to the first neighbor of $z_{f(m)}$ in the sequence $w_1, w_2, \dots, w_s$. Note that all four charged vertices were previously uncharged.

- Suppose next that $d(x) + d(y) = 3$. Assume that $d(x) > d(y)$, as the case $d(x) < d(y)$ can be handled symmetrically by mirroring the embedding of G . We need to further brake down this case into three subcases.
 - First, suppose that $d(z_1) = 2$ (see Figure 8.10a). We define a set $\mathcal{I}_L$ by means of the basic algorithm. We prove that $\mathcal{I}_L$ satisfies Property C (if $d(x) < d(y)$ and $d(z_k) = 2$, rather than $d(x) > d(y)$ and $d(z_1) = 2$, we can define a set $\mathcal{I}_L$ that satisfies Property B). In order to prove that $G[\mathcal{I}_L \cup \{y\}]$ is outerplane, we observe that the neighbor x of the vertices $x_1 = y_1, x_2$ is not in $\mathcal{I}_L$. In order to prove that $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, we charge x to $x_1 = y_1 = y_{d(y)}$ and to $x_2 = x_{d(x)}$. Note that, while the basic algorithm always guarantees that $y_{d(y)}$ is not charged with any vertex z_j , the fact that $x_{d(x)}$ is not charged with any vertex z_j relies on the assumption $d(z_1) = 2$.
 - Second, suppose that $d(z_1) \geq 3$ and m is even (see Figure 8.10b). We define a set $\mathcal{I}_L$ by means of the basic algorithm. We prove that $\mathcal{I}_L$ satisfies Property D. In order to prove that $G[\mathcal{I}_L \cup \{x, y\}]$ is outerplane, we observe that the neighbor z_1 of the vertex x_2 is not in $\mathcal{I}_L$, given that $d(z_1) \geq 3$. Also, the neighbor z_k of the vertex $x_1 = y_1$ is not in $\mathcal{I}_L \cup \{x, y\}$; this is true either because $d(z_k) \geq 3$, or

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

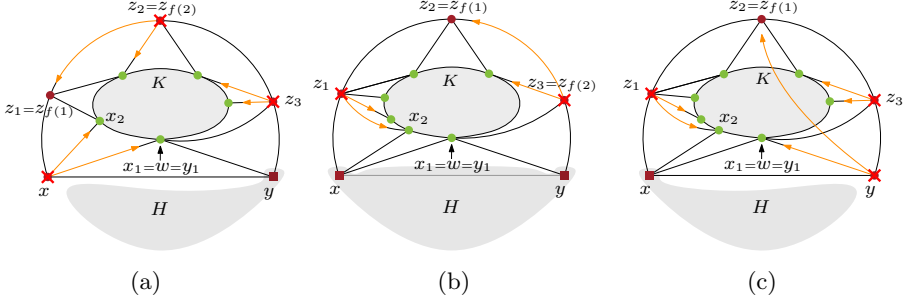


Figure 8.10: Illustrations for Case 4 when $d(x) + d(y) = 3$ (a) if $d(z_1) = 2$, (b) if $d(z_1) \geq 3$ and m is even, and (c) if $d(z_1) \geq 3$ and m is odd.

because z_k coincides with $z_{f(m)}$ and m is even, by assumption. Since in this case we do not need to charge x and y , the bound $|\mathcal{I}_L| \geq 2(n_L - 2)/3$ directly follows.

- *Third, suppose that $d(z_1) \geq 3$ and m is odd* (see Figure 8.10c). We define a set $\mathcal{I}_L$ by means of the basic algorithm. We prove that $\mathcal{I}_L$ satisfies Property B (if $d(x) = 1 < d(y)$, $d(z_k) \geq 3$, and m is odd, rather than $d(x) > d(y)$, $d(z_1) \geq 3$, and m is odd, we can define a set $\mathcal{I}_L$ that satisfies Property C). In order to prove that $G[\mathcal{I}_L \cup \{x\}]$ is outerplane, we observe that the neighbor z_1 of the vertex x_2 is not in $\mathcal{I}_L$, given that $d(z_1) \geq 3$, and that the neighbor y of the vertex $x_1 = y_1$ is not in $\mathcal{I}_L \cup \{x\}$. In order to prove that $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, we charge y to $x_1 = y_1 = y_{d(y)}$ and to $z_{f(m)}$. Note that both such vertices were previously uncharged.
- *Suppose finally that $d(x) + d(y) = 2$* (see Figure 8.11). Hence, x and y are neighbors of w and of no other vertex of K . We define a set $\mathcal{I}_L$ by means of the basic algorithm, however if m is odd and greater than 1, we remove $z_{f(m)}$ from $\mathcal{I}_L$. We prove that $\mathcal{I}_L$ satisfies Property D. In order to prove that $G[\mathcal{I}_L \cup \{x, y\}]$ is outerplane, we need to show that there exists a neighbor of $x_1 = y_1 = w$ among $y, x, z_1, \dots, z_k$ that is not in $\mathcal{I}_L \cup \{x, y\}$. Clearly, such a neighbor cannot be x or y , since these vertices are in $\mathcal{I}_L \cup \{x, y\}$. Note that z_1 and z_k are neighbors of w and in this case they are guaranteed to be distinct vertices, given that L is internally-triangulated, that $d(x) = d(y) = 1$, and that K is not a single vertex. If z_1 or z_k has internal degree at least 3, then that vertex is the

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

desired neighbor of w not in $\mathcal{I}_L$. Otherwise, z_1 and z_k have both internal degree 2, thus $m \geq 2$, and then $z_k = z_{f(m)}$ is not in $\mathcal{I}_L$. In order to prove that $|\mathcal{I}_L| \geq 2(n_L - 2)/3$, we do not need to charge x and y ; however, if m is odd and greater than 1, then we need to charge $z_{f(m)}$. We charge it to its first neighbor in the sequence $w_1, w_2, \dots, w_s$ and to the first neighbor of $z_{f(1)}$ in the sequence $w_1, w_2, \dots, w_s$. Note that both such vertices were previously uncharged.

This completes the proof of the lemma. $\square$

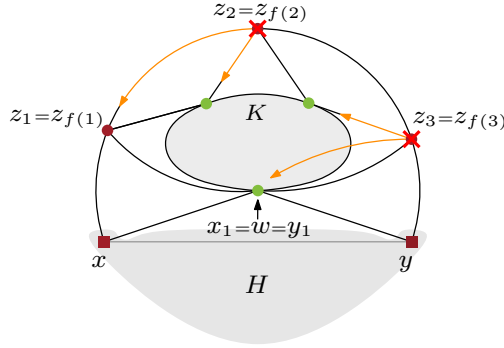


Figure 8.11: Illustration for Case 4 when $d(x) + d(y) = 2$.

We can now apply induction as follows. Let $\mathcal{I}_L$ be a set as in [Lemma 8.9](#). If $\mathcal{I}_L$ satisfies Property A, let H be the subgraph of G induced by $V(G) - V(L)$; otherwise, if it satisfies Property B, let H be the subgraph of G induced by $V(G) - (V(L) - \{x\})$; otherwise, if it satisfies Property C, let H be the subgraph of G induced by $V(G) - (V(L) - \{y\})$; otherwise, if it satisfies Property D, let H be the subgraph of G induced by $V(G) - (V(L) - \{x, y\})$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \mathcal{I}_L$. We prove that $\mathcal{I}$ is a good set. Note that $|\mathcal{I}| = |\mathcal{I}_H| + |\mathcal{I}_L|$. If $\mathcal{I}_L$ satisfies Property A, then $|V(H)| = n - n_L$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - n_L)/3$ and, by [Lemma 8.9](#), we have $|\mathcal{I}_L| \geq 2n_L/3$, thus $|\mathcal{I}| \geq 2n/3$. If $\mathcal{I}_L$ satisfies Property B or Property C, then $|V(H)| = n - n_L + 1$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - n_L + 1)/3$ and, by [Lemma 8.9](#), we have $|\mathcal{I}_L| \geq 2(n_L - 1)/3$, thus $|\mathcal{I}| \geq 2n/3$. Finally, if $\mathcal{I}_L$ satisfies Property D, then $|V(H)| = n - n_L + 2$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - n_L + 2)/3$ and, by [Lemma 8.9](#), we have $|\mathcal{I}_L| \geq 2(n_L - 2)/3$, thus $|\mathcal{I}| \geq 2n/3$. In order to prove that $G[\mathcal{I}]$ is outerplane, note that $G[\mathcal{I}_H]$ is

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

outerplane by induction. If $\mathcal{I}_L$ satisfies Property A, then $G[\mathcal{I}_L]$ is outerplane by [Lemma 8.9](#). Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_L]$ are each in the outer face of the other one and share no vertex, by [Observation 8.2](#) we have that $G[\mathcal{I}]$ is an outerplane graph. If $\mathcal{I}_L$ satisfies Property B, then $G[\mathcal{I}_L \cup (\{x\} \cap \mathcal{I}_H)]$ is outerplane by [Lemma 8.9](#). Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_L \cup (\{x\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one and share either no vertex or a single vertex, by [Observation 8.2](#) we have that $G[\mathcal{I}]$ is an outerplane graph. Similarly, if $\mathcal{I}_L$ satisfies Property C, then $G[\mathcal{I}_L \cup (\{y\} \cap \mathcal{I}_H)]$ is outerplane by [Lemma 8.9](#). Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_L \cup (\{y\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one and share either no vertex or a single vertex, by [Observation 8.2](#) we have that $G[\mathcal{I}]$ is an outerplane graph. Finally, if $\mathcal{I}_L$ satisfies Property D, then $G[\mathcal{I}_L \cup (\{x, y\} \cap \mathcal{I}_H)]$ is outerplane by [Lemma 8.9](#). Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_L \cup (\{x, y\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#) we have that $G[\mathcal{I}]$ is an outerplane graph.

In the remaining cases, K is not biconnected. Let B be an extremal leaf of K . Suppose first that B is a non-trivial block.

Case 5: B is non-trivial and cushy. Let $n_B = |V(G_B)|$, where G_B is the cage graph of B . Let $c_B, w_1 = u_B, w_2, \dots, w_{s-1}, w_s = v_B$ be the clockwise order of the vertices along the outer face of B , and let $c_B, z_1 = \ell_B, z_2, \dots, z_{k-1}, z_k = r_B$ be the clockwise order of the vertices along the outer face of G_B . The *internal degree* $d(z_i)$ of a vertex $z_i \neq c_B$ on the boundary of G_B is the number of its neighbors in B . Note that, since Cases 1 and 2 do not apply, the internal degree of every vertex among $z_2, \dots, z_{k-1}$ is at least two. The internal degree of z_1 and z_k is also at least two. Indeed, z_1 and z_k are both neighbors of c_B . Also, $c_B \ell_B$ is the edge of G that precedes $c_B u_B$ in clockwise direction, hence the edge $\ell_B u_B$ exists, since G is internally-triangulated. Likewise, the edge $r_B v_B$ exists. We further distinguish two cases.

In **Case 5.1**, $d(z_1) = d(z_k) = 2$ and either (i) $k = 3$ and $d(z_2) = 2$ (see [Figure 8.12a](#)), or (ii) $k \geq 3$ and $d(z_j) = 3$, for $j = 2, \dots, k-1$ (see [Figure 8.12b](#)). Note that, if (i) holds, then $k = 3$, $s = 2$, and $n_B = 6$, while if (ii) holds, then $s = 2(k-2) + 1 = 2k-3$ and $n_B = 3(k-2) + 4 = 3k-2$.

Suppose first that the edge $\ell_B r_B$ does not belong to G (refer to [Figure 8.12](#)). We construct a 2-outerplane graph H from G by removing the vertices in $V(G_B) - \{c_B, \ell_B, r_B\}$ and their incident edges, and by adding the edge $\ell_B r_B$ in the outer face. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup (V(B) - \{c_B\})$. Note that c_B, ℓ_B , and r_B belong to $\mathcal{I}$ if and only if they belong to $\mathcal{I}_H$. We prove that $\mathcal{I}$

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

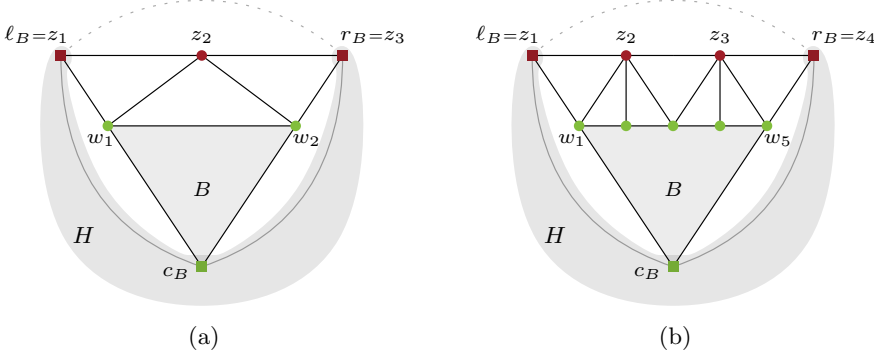


Figure 8.12: Illustrations for Case 5.1 when the edge $\ell_B r_B$ does not belong to G (a) if $k = 3$ and $d(z_2) = 2$, and (b) if $k \geq 3$ and $d(z_j) = 3$, for $j = 2, \dots, k-1$.

is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + s$ and $|V(H)| = n - n_B + 3$. If (i) holds, then $|\mathcal{I}| = |\mathcal{I}_H| + 2 \geq 2(n-3)/3 + 2 = 2n/3$. If (ii) holds, then $|\mathcal{I}| = |\mathcal{I}_H| + 2(k-2) + 1 \geq 2(n-3(k-2)-1)/3 + 2(k-2) + 1 > 2n/3$. Suppose, for a contradiction, that $G[\mathcal{I}]$ contains a cycle $\mathcal{C}$ which contains a vertex $w \in \mathcal{I}$ in its interior. Note that $w \notin \{w_1, \dots, w_s\}$, since $w_1, \dots, w_s$ are incident to the outer face of $G[\mathcal{I}]$, hence $w \in \mathcal{I}_H$. Also, $\mathcal{C}$ uses vertices among $w_1, \dots, w_s$, as otherwise $G[\mathcal{I}_H]$ would also contain $\mathcal{C}$, and $\mathcal{I}_H$ would not be a good set. Since any cycle in G on the vertices $V(B) \cup \{\ell_B, r_B\}$ does not include any vertex in its interior, it follows that $\mathcal{C}$ also contains vertices of $V(H) - \{\ell_B, r_B\}$. Let then $\mathcal{P}$ be the maximal path in $\mathcal{C}$ on the vertices in $V(B) \cup \{\ell_B, r_B\}$ and note that the end-vertices of $\mathcal{P}$ are two among c_B, ℓ_B, r_B . Thus, $\mathcal{P}$ can be replaced by the edge between its end-vertices, resulting in a cycle of H which also contains w in its interior and whose vertices all belong to $\mathcal{I}_H$, which implies that $\mathcal{I}_H$ is not a good set, a contradiction.

Suppose next that the edge $\ell_B r_B$ belongs to G . Let b be the number of vertices inside the cycle $c_B \ell_B r_B$. Note that $b \geq 1$, given that K is not biconnected. Let G'_B be the subgraph of G induced by the vertices inside or on the boundary of the cycle $C'_B := z_1, z_2, \dots, z_{k-1}, z_k$. Observe that G'_B consists of C'_B and of K , which is inside C'_B . Hence, $|V(G'_B)| = n_B + b$.

- If $b \geq 3$ (refer to [Figure 8.13](#)), let H be the subgraph of G induced by $V(G) - V(G'_B)$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup V(K)$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(K)|$ and $|V(H)| = n - |V(G'_B)|$. If (i) holds

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

(see Figure 8.13a), then we have $n_B = 6$ and, by the assumption $b \geq 3$, we have $|V(G'_B)| \geq 9$. It follows that $|\mathcal{I}| = |\mathcal{I}_H| + |V(G'_B)| - 3 \geq 2(n - |V(G'_B)|)/3 + |V(G'_B)| - 3 = 2n/3 + |V(G'_B)|/3 - 3 \geq 2n/3$. If (ii) holds (see Figure 8.13b), then, by the assumption $b \geq 3$, we have $|V(G'_B)| \geq 3k + 1$. It follows that $|\mathcal{I}| = |\mathcal{I}_H| + |V(G'_B)| - k \geq 2(n - |V(G'_B)|)/3 + |V(G'_B)| - k = 2n/3 + |V(G'_B)|/3 - k > 2n/3$. Also, $G[\mathcal{I}_H]$ is outerplane by induction and $K = G[V(K)]$ is outerplane since it is a component of $G[L_2]$. Since $G[\mathcal{I}_H]$ and K are each in the outer face of the other one, and share no vertex, by Observation 8.2, we have that $G[\mathcal{I}]$ is an outerplane graph.

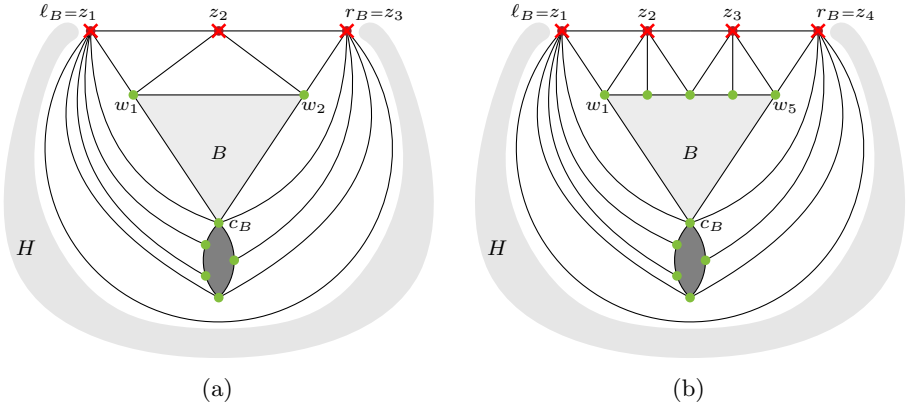


Figure 8.13: Illustrations for Case 5.1 when the edge $\ell_B r_B$ belongs to G and $b \geq 3$ (a) if $k = 3$ and $d(z_2) = 2$, and (b) if $k \geq 3$ and $d(z_j) = 3$, for $j = 2, \dots, k - 1$.

- If $b = 1$ or $b = 2$ (refer to Figure 8.14), let U be the set of vertices inside $c_B \ell_B r_B$; we have $U = \{u_1\}$ if $b = 1$ and $U = \{u_1, u_2\}$ if $b = 2$. Since G is internally-triangulated, there exists one of ℓ_B and r_B , possibly both, that is adjacent to all the vertices in U . Assume that ℓ_B is adjacent to all the vertices in U , as the other case is similar. Let H be the subgraph of G induced by $V(G) - (V(G'_B) - \{\ell_B\})$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup V(K)$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(K)|$ and $|V(H)| = n - |V(G'_B)| + 1$. If (i) holds (see Figure 8.14a), then we have $n_B = 6$ and $|V(G'_B)| \geq 7$. It follows that $|\mathcal{I}| = |\mathcal{I}_H| + |V(G'_B)| - 3 \geq 2(n - |V(G'_B)| + 1)/3 + |V(G'_B)| - 3 = 2n/3 + |V(G'_B)|/3 - 7/3 \geq 2n/3$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

If (ii) holds (see [Figure 8.14b](#)), then we have $|V(G'_B)| \geq 3k - 1$. It follows that $|\mathcal{I}| = |\mathcal{I}_H| + |V(G'_B)| - k \geq 2(n - |V(G'_B)| + 1)/3 + |V(G'_B)| - k = 2n/3 + |V(G'_B)|/3 - k + 2/3 > 2n/3$. Also, $G[\mathcal{I}_H]$ is outerplane by induction and $G[V(K) \cup (\mathcal{I}_H \cap \{r_B\})]$ is outerplane since each vertex of K is adjacent to a vertex among $z_1 = \ell_B, z_2, \dots, z_{k-1}$; note that all such vertices do not belong to $\mathcal{I}$. Since $G[\mathcal{I}_H]$ and $G[V(K) \cup (\mathcal{I}_H \cap \{r_B\})]$ are each in the outer face of the other one, and share either no vertex or a single vertex, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

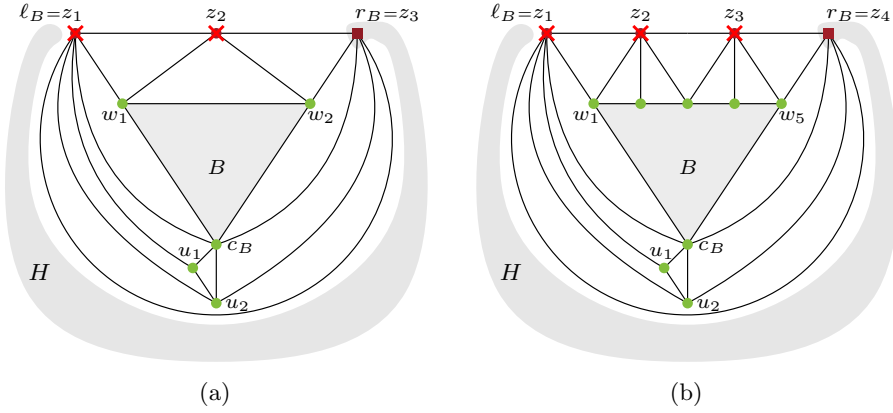


Figure 8.14: Illustrations for Case 5.1 when the edge $\ell_B r_B$ belongs to G and $b = 1$ or $b = 2$ (a) if $k = 3$ and $d(z_2) = 2$, and (b) if $k \geq 3$ and $d(z_j) = 3$, for $j = 2, \dots, k - 1$.

In **Case 5.2**, we have that Case 5.1 does not apply. Although the setting is different, the general strategy we employ to handle this case is similar to the one of Case 4. We find a set $\mathcal{I}_B$ of vertices in G_B so that excluding or disregarding the vertices that G_B shares with the rest of the graph (these vertices are c_B , ℓ_B , and r_B , which play a role similar to the one that in Case 4 is played by x and y) we can ensure, on one hand, that the number of vertices in $\mathcal{I}_B$ is at least twice the number of vertices of G_B that are excluded from $\mathcal{I}_B$ and, on the other hand, that $\mathcal{I}_B$, together with the set of vertices that are inductively selected in the rest of the graph, induces an outerplane graph. Formally, we prove the following.

Lemma 8.10. *There exists a set $\mathcal{I}_B \subset V(G_B)$ such that $\{c_B, \ell_B, r_B\} \cap \mathcal{I}_B = \emptyset$ and such that (at least) one of the following properties is satisfied:*

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

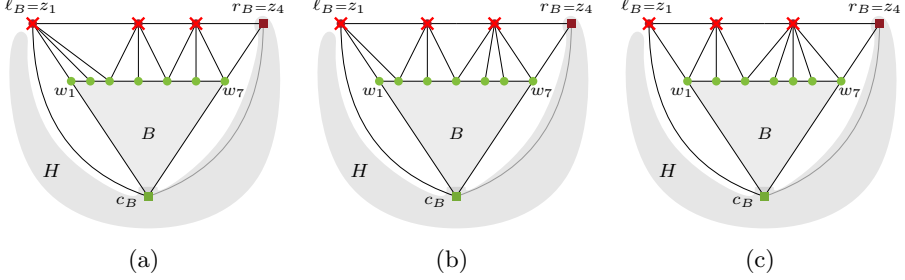


Figure 8.15: Illustrations for Case 5.2.1 (a) if (i) holds, (b) if (ii) holds, and (c) if (iii) holds.

- Property A: $|\mathcal{I}_B| \geq 2(n_B - 1)/3$ and $G[\mathcal{I}_B \cup \{c_B\}]$ is an outerplane graph;
- Property B: $|\mathcal{I}_B| \geq 2(n_B - 2)/3$ and $G[\mathcal{I}_B \cup \{c_B, \ell_B\}]$ is an outerplane graph; and
- Property C: $|\mathcal{I}_B| \geq 2(n_B - 2)/3$ and $G[\mathcal{I}_B \cup \{c_B, r_B\}]$ is an outerplane graph.

Proof. Let $z_{f(1)}, z_{f(2)}, \dots, z_{f(m)}$ be the sequence $z_2, \dots, z_{k-1}$ restricted to the $m \geq 0$ vertices with internal degree 2; observe that, differently from the proof of [Lemma 8.9](#), the vertices z_1 and z_k are not in the sequence $z_{f(1)}, z_{f(2)}, \dots, z_{f(m)}$.

The set $\mathcal{I}_B$ is defined according to the following rules. First, all the vertices of B , except for c_B , are in $\mathcal{I}_B$. Second, ℓ_B and r_B are not in $\mathcal{I}_B$. Third, each vertex z_j with $d(z_j) \geq 3$ is not in $\mathcal{I}_B$. Finally, a vertex $z_{f(j)}$ with internal degree 2 is in $\mathcal{I}_B$ if j is odd and not in $\mathcal{I}_B$ if j is even. Note that $\{c_B, \ell_B, r_B\} \cap \mathcal{I}_B = \emptyset$, as required. We now distinguish three cases.

In **Case 5.2.1**, we have $d(z_k) = 2$ and at least one of the following conditions is satisfied:

- (i) $3 \leq d(z_1) \leq 4$ and $d(z_2) = d(z_3) = \dots = d(z_{k-1}) = 3$ (see [Figure 8.15a](#));
- (ii) $d(z_1) \leq 3$, $d(z_i) = 4$, and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$ (see [Figure 8.15b](#));
- (iii) $d(z_1) = 2$, $d(z_i) = 5$, and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$ (see [Figure 8.15c](#));

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

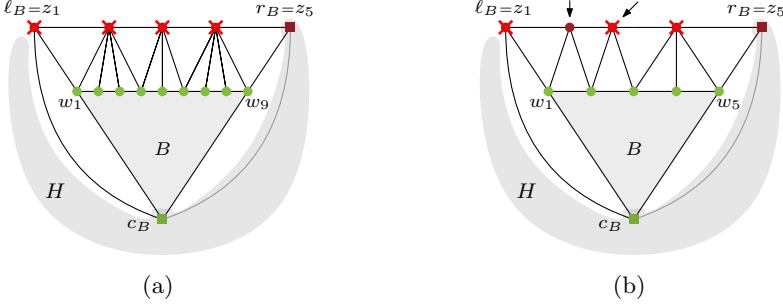


Figure 8.16: Illustrations for Case 5.2.1 (a) if (iv) holds and (b) if (v) holds.

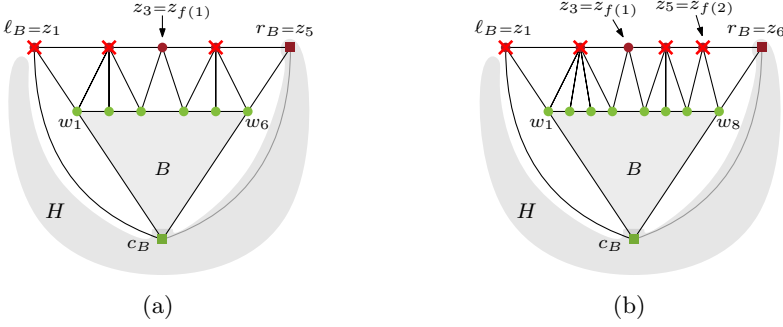


Figure 8.17: Illustrations for Case 5.2.1 (a) if (vi) holds and (b) if (vii) holds.

- (iv) $d(z_1) = 2$, $d(z_i) = 4$, $d(z_{i'}) = 4$, and $d(z_j) = 3$, for some distinct $i, i' \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$ and $j \neq i'$ (see Figure 8.16a);
- (v) $m = 2$ and $d(z_j) \leq 3$ for $j = 1, \dots, k-1$, or $m = 4$, $d(z_1) = 2$, and $d(z_j) \leq 3$ for $j = 2, \dots, k-1$ (see Figure 8.16b);
- (vi) $m = 1$, $d(z_1) = 2$, $d(z_{k-1}) = 3$, and $d(z_j) \leq 3$ for $j = 2, \dots, k-2$ (see Figure 8.17a); or
- (vii) $d(z_1) = 2$, $m = 2$, $d(z_i) = 4$, and $d(z_j) \leq 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$ (see Figure 8.17b).

In this case, we have that $\mathcal{I}_B$ satisfies Property C.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

In order to prove that $G[\mathcal{I}_B \cup \{c_B, r_B\}]$ is an outerplane graph, it suffices to show that, for each vertex in $V(B) - \{c_B\}$, there exists a neighbor among $z_1, \dots, z_k$ which is not in $\mathcal{I}_B \cup \{c_B, r_B\}$. This is done similarly to the proof of [Lemma 8.9](#). Namely, consider a vertex w_i , for some $2 \leq i \leq s-1$. If w_i is a neighbor of a vertex z_j with $d(z_j) \geq 3$, then z_j is not in $\mathcal{I}_B \cup \{c_B, r_B\}$, hence it is the desired neighbor of w_i . Each remaining vertex w_i with $2 \leq i \leq s-1$ is only neighbor of vertices z_j with $d(z_j) = 2$ and with $2 \leq j \leq k-1$; indeed, w_i is not a neighbor of z_k , given that $d(z_k) = 2$ and $i \leq s-1$, and if w_i were a neighbor of z_1 , then $d(z_1) \geq 3$, given that $i \geq 2$. Since G_B is internally-triangulated, it follows that w_i is a neighbor of two vertices $z_{f(j)}$ and $z_{f(j+1)}$ with $d(z_{f(j)}) = d(z_{f(j+1)}) = 2$ that are consecutive along the boundary of G_B . By construction, one of them is not in $\mathcal{I}_B$, hence it is the desired neighbor of w_i . Furthermore, w_1 is adjacent to $z_1 = \ell_B$, which is not in $\mathcal{I}_B \cup \{c_B, r_B\}$. Finally, w_s is adjacent to z_{k-1} , since $d(z_k) = 2$, and z_{k-1} is not in $\mathcal{I}_B \cup \{c_B, r_B\}$. Indeed, if $k = 2$, then $z_{k-1} = z_1$ is not in $\mathcal{I}_B \cup \{c_B, r_B\}$ by construction. If $k \geq 3$ and $d(z_{k-1}) \geq 3$, then z_{k-1} is not in $\mathcal{I}_B$ by construction. That $d(z_{k-1}) = 2$ can only happen in cases (v) or (vii), where we have $m = 2$ or $m = 4$. Then z_{k-1} is not in $\mathcal{I}_B$ since a vertex $z_{f(j)}$ with j even and with $d(z_{f(j)}) = 2$ is not in $\mathcal{I}_B$.

We now prove that $|\mathcal{I}_B| \geq 2(n_B - 2)/3$.

- (i) If $d(z_1) = d(z_2) = d(z_3) = \dots = d(z_{k-1}) = 3$, then $|\mathcal{I}_B| = s = 2k - 2$ and $n_B = 3k - 1$. Hence, we have that $|\mathcal{I}_B| = 2k - 2 = 2((3k - 1) - 2)/3 = 2(n_B - 2)/3$. If $d(z_1) = 4$ and $d(z_2) = d(z_3) = \dots = d(z_{k-1}) = 3$, then $|\mathcal{I}_B| = s = 2k - 1$ and $n_B = 3k$. Hence, we have that $|\mathcal{I}_B| = 2k - 1 > 2(3k - 2)/3 = 2(n_B - 2)/3$.
- (ii) If $d(z_1) = 2$, $d(z_i) = 4$, and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$, then $|\mathcal{I}_B| = s = 2k - 2$ and $n_B = 3k - 1$. Hence, we have that $|\mathcal{I}_B| = 2k - 2 = 2((3k - 1) - 2)/3 = 2(n_B - 2)/3$. If $d(z_1) = 3$, $d(z_i) = 4$, and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$, then $|\mathcal{I}_B| = s = 2k - 1$ and $n_B = 3k$. Hence, we have that $|\mathcal{I}_B| = 2k - 1 > 2(3k - 2)/3 = 2(n_B - 2)/3$.
- (iii) If $d(z_1) = 2$, $d(z_i) = 5$, and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$, then $|\mathcal{I}_B| = s = 2k - 1$ and $n_B = 3k$. Hence, we have that $|\mathcal{I}_B| = 2k - 1 > 2(3k - 2)/3 = 2(n_B - 2)/3$.
- (iv) If $d(z_1) = 2$, $d(z_i) = 4$, $d(z_{i'}) = 4$, and $d(z_j) = 3$, for some distinct $i, i' \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$ and $j \neq i'$, then $|\mathcal{I}_B| = s = 2k - 1$ and $n_B = 3k$. Hence, we have that $|\mathcal{I}_B| = 2k - 1 > 2(3k - 2)/3 = 2(n_B - 2)/3$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

- (v) If $m = 2$, $d(z_1) = 2$, and $d(z_j) \leq 3$ for $j = 2, \dots, k-1$, then $|\mathcal{I}_B| = s+1 = 2k-4$ and $n_B = 3k-4$. Hence, we have that $|\mathcal{I}_B| = 2k-4 = 2((3k-4)-2)/3 = 2(n_B-2)/3$. If $m = 2$, $d(z_1) = 3$, and $d(z_j) \leq 3$ for $j = 2, \dots, k-1$, then $|\mathcal{I}_B| = s+1 = 2k-3$ and $n_B = 3k-3$. Hence, we have that $|\mathcal{I}_B| = 2k-3 > 2((3k-3)-2)/3 = 2(n_B-2)/3$. If $m = 4$, $d(z_1) = 2$, and $d(z_j) \leq 3$ for $j = 2, \dots, k-1$, then $|\mathcal{I}_B| = s+2 = 2k-5$ and $n_B = 3k-6$. Hence, we have that $|\mathcal{I}_B| = 2k-5 > 2((3k-6)-2)/3 = 2(n_B-2)/3$.
- (vi) If $m = 1$, $d(z_1) = 2$, $d(z_{k-1}) = 3$, and $d(z_j) \leq 3$ for $j = 2, \dots, k-2$, then $|\mathcal{I}_B| = s+1 = 2k-3$ and $n_B = 3k-3$. Hence, we have that $|\mathcal{I}_B| = 2k-3 > 2((3k-3)-2)/3 = 2(n_B-2)/3$.
- (vii) If $d(z_1) = 2$, $m = 2$, $d(z_i) = 4$, and $d(z_j) \leq 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$, then $|\mathcal{I}_B| = s+1 = 2k-3$ and $n_B = 3k-3$. Hence, we have that $|\mathcal{I}_B| = 2k-3 > 2((3k-3)-2)/3 = 2(n_B-2)/3$.

In **Case 5.2.2**, we have $d(z_1) = 2$ and at least one of the following conditions is satisfied:

- (i) $3 \leq d(z_k) \leq 4$ and $d(z_2) = d(z_3) = \dots = d(z_{k-1}) = 3$;
- (ii) $d(z_k) \leq 3$, $d(z_i) = 4$, and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$;
- (iii) $d(z_k) = 2$, $d(z_i) = 5$, and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$;
- (iv) $d(z_k) = 2$, $d(z_i) = 4$, $d(z_{i'}) = 4$, and $d(z_j) = 3$, for some distinct $i, i' \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$ and $j \neq i'$;
- (v) $m = 2$ and $d(z_j) \leq 3$ for $j = 2, \dots, k$, or $m = 4$, $d(z_k) = 2$, and $d(z_j) \leq 3$ for $j = 2, \dots, k-1$;
- (vi) $m = 1$, $d(z_k) = 2$, $d(z_2) = 3$, and $d(z_j) \leq 3$ for $j = 3, \dots, k-1$; or
- (vii) $d(z_k) = 2$, $m = 2$, $d(z_i) = 4$, and $d(z_j) \leq 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$.

In this case, we have that $\mathcal{I}_B$ satisfies Property B. The proof of this statement is symmetric to Case 5.2.1. Note that Cases 5.2.1 and 5.2.2 might hold both true for G .

If we are neither in Case 5.2.1 nor in Case 5.2.2, then we are in **Case 5.2.3** and we have that $\mathcal{I}_B$ satisfies Property A. The proof that $G[\mathcal{I}_B \cup \{c_B\}]$ is

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

an outerplane graph is similar, and actually simpler, to Case 5.2.1, since the neighbors z_1 and z_k of w_1 and w_s , respectively, are now not in $\mathcal{I}_B \cup \{c_B\}$.

The proof that $|\mathcal{I}_B| \geq 2(n_B - 1)/3$ is similar in spirit to, however more complex than, the proof of [Lemma 8.9](#). We charge each vertex of G_B not in $\mathcal{I}_B$ and different from c_B to two vertices in $\mathcal{I}_B$, so that each vertex in $\mathcal{I}_B$ is charged with at most one vertex.

The charging scheme is defined iteratively on the vertices $z_1, \dots, z_k$ that do not belong to $\mathcal{I}_B$. Initially, we charge z_1 as follows. If $d(z_1) \geq 3$, then z_1 is charged to w_1, w_2 , while if $d(z_1) = 2$, then z_1 is charged to w_1 (and one more charged vertex will be defined later). For any $j = 1, \dots, k$, let $\mathcal{I}_B^j$ denote the subset of $\mathcal{I}_B$ which comprises all the vertices $z_i \in \mathcal{I}_B$ with $i \leq j$ and all the vertices $w_1, w_2, \dots, w_{g(j)}$, where $w_{g(j)}$ is the last neighbor of z_j in the sequence $w_1, w_2, \dots, w_s$. After having charged the vertices among $z_1, z_2, \dots, z_j$ that are not in $\mathcal{I}_B^j$ to vertices in $\mathcal{I}_B^j$, we have a *stock of charge* γ_j , which is an integer that represents the number of vertices in $\mathcal{I}_B^j$ that have not been charged with any vertex among $z_1, z_2, \dots, z_j$. Initially, this is defined as $\gamma_1 = d(z_1) - 3$. Indeed, $\mathcal{I}_B^1 = \{w_1, \dots, w_{d(z_1)-1}\}$ (note that c_B contributes to $d(z_1)$, however it is not among $w_1, w_2, \dots, w_s$). Hence, if $d(z_1) \geq 3$, then γ_j correctly represents the fact that z_1 has been charged to w_1, w_2 and the vertices $w_3, w_4, \dots, w_{g(j)}$ have not been charged with z_1 . If $d(z_1) = 2$, then $\gamma_j = -1$, which represents the fact that z_1 needs to be charged to one additional vertex.

We now prove that, for any $j \in \{1, \dots, k-1\}$, we can design a charging scheme that satisfies the following properties. First, the vertices $z_1, \dots, z_j$ have been charged to two vertices in $\mathcal{I}_B^j$, with the possible exception of z_1 which might have been charged to w_1 only, so that each vertex in $\mathcal{I}_B^j$ has been charged with at most one vertex. Second, if $d(z_1) \geq 3$ or if $d(z_i) \neq 3$, for some $2 \leq i \leq j$, then z_1 has also been charged to two vertices in $\mathcal{I}_B^j$. Third, let m_j be equal to the number of vertices z_i among $z_2, z_3, \dots, z_j$ such that $d(z_i) = 2$. Then the stock of charge γ_j is equal to the sum S_j of the following terms:

- $d(z_1) - 3$;
- $d(z_i) - 3$, for each vertex z_i with $d(z_i) \geq 3$ and with $2 \leq i \leq j$;
- $\lfloor m_j/2 \rfloor$; and
- 2, if m_j is odd.

Observe that $S_j \geq -1$, and that $S_j = -1$ if and only if $d(z_1) = 2$ and $d(z_i) = 3$, for $i = 2, \dots, j$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

The initial charge of z_1 trivially satisfies these properties; note that $m_1 = 0$. Suppose now that the charging scheme for $z_1, \dots, z_j$ satisfies these properties, for some $j \in \{1, \dots, k-2\}$. We prove that the same properties are satisfied after considering z_{j+1} .

- Suppose first that $d(z_{j+1}) = 3$. Then $|\mathcal{I}_B^{j+1}| = |\mathcal{I}_B^j| + 2$, since $w_{g(j+1)-1}$ and $w_{g(j+1)}$ belong to $\mathcal{I}_B^{j+1}$ and not to $\mathcal{I}_B^j$; note that $w_{g(j+1)-2}$ is also a neighbor of z_j , hence it is already in $\mathcal{I}_B^j$. Hence, we can charge z_{j+1} to two uncharged vertices in $\mathcal{I}_B^{j+1}$ and all the required properties remain satisfied. In particular, z_1 has been charged to two vertices in $\mathcal{I}_B^{j+1}$ if and only if it has been charged to two vertices in $\mathcal{I}_B^j$, given that $d(z_i) \neq 3$ for some $2 \leq i \leq j+1$ if and only if $d(z_i) \neq 3$ for some $2 \leq i \leq j$. Also, $\gamma_{j+1} = \gamma_j$, given that $\mathcal{I}_B^{j+1}$ contains two vertices more than $\mathcal{I}_B^j$ and that z_{j+1} is charged to two vertices in $\mathcal{I}_B^{j+1}$. Note that $S_{j+1} = S_j$, given that $m_{j+1} = m_j$ and that $d(z_{j+1}) - 3 = 0$. Since $\gamma_j = S_j$, by hypothesis, we have that $\gamma_{j+1} = S_{j+1}$.
- Suppose next that $d(z_{j+1}) > 3$. Then $|\mathcal{I}_B^{j+1}| = |\mathcal{I}_B^j| + d(z_{j+1}) - 1 > |\mathcal{I}_B^j| + 2$. Hence, after charging z_{j+1} to two uncharged vertices in $\mathcal{I}_B^{j+1}$, there is at least one more uncharged vertex in $\mathcal{I}_B^{j+1}$ than in $\mathcal{I}_B^j$. If $\gamma_j = -1$, then we also charge z_1 to an uncharged vertex in $\mathcal{I}_B^{j+1}$, and all the required properties remain satisfied. In particular, z_1 has now been charged to two vertices in $\mathcal{I}_B^{j+1}$. Also, $\gamma_{j+1} = \gamma_j + d(z_{j+1}) - 3$, given that $\mathcal{I}_B^{j+1}$ contains $d(z_{j+1}) - 1$ vertices more than $\mathcal{I}_B^j$ and that z_{j+1} is charged to two vertices in $\mathcal{I}_B^{j+1}$; the fact that $\gamma_j = -1$ if z_1 has only been charged to one vertex of $\mathcal{I}_B^j$ makes the equation $\gamma_{j+1} = \gamma_j + d(z_{j+1}) - 3$ valid even if we require a third vertex of $\mathcal{I}_B^{j+1}$ to be charged with z_1 . Note that $S_{j+1} = S_j + d(z_{j+1}) - 3$, given that $m_{j+1} = m_j$. Since $\gamma_j = S_j$, by hypothesis, we have that $\gamma_{j+1} = S_{j+1}$.
- Suppose next that $d(z_{j+1}) = 2$ and that m_{j+1} is odd. Then $|\mathcal{I}_B^{j+1}| = |\mathcal{I}_B^j| + 2$, since z_{j+1} and $w_{g(j+1)}$ belong to $\mathcal{I}_B^{j+1}$ and not to $\mathcal{I}_B^j$. Note that, in this case, we do not need to charge z_{j+1} to any vertex, since z_{j+1} belongs to $\mathcal{I}_B$. However, if $\gamma_j = -1$, then we also charge z_1 to an uncharged vertex in $\mathcal{I}_B^{j+1}$, and all the required properties remain satisfied. In particular, z_1 has now been charged to two vertices in $\mathcal{I}_B^{j+1}$. Also, $\gamma_{j+1} = \gamma_j + 2$, given that $\mathcal{I}_B^{j+1}$ contains 2 vertices more than $\mathcal{I}_B^j$; the fact that $\gamma_j = -1$ if z_1 has

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

only been charged to one vertex of $\mathcal{I}_B^j$ makes the equation $\gamma_{j+1} = \gamma_j + 2$ valid even if we require a vertex of $\mathcal{I}_B^{j+1}$ to be charged with z_1 . Note that $S_{j+1} = S_j + 2$, given that $\lfloor m_{j+1}/2 \rfloor = \lfloor m_j/2 \rfloor$ and the 2 additive term is present for S_{j+1} , given that m_{j+1} is odd, but not for S_j , given that m_j is even. Since $\gamma_j = S_j$, by hypothesis, we have that $\gamma_{j+1} = S_{j+1}$.

- Suppose finally that $d(z_{j+1}) = 2$ and that m_{j+1} is even. Then $|\mathcal{I}_B^{j+1}| = |\mathcal{I}_B^j| + 1$, since $w_{g(j+1)} + 1$ belongs to $\mathcal{I}_B^{j+1}$ and not to $\mathcal{I}_B^j$. Since m_{j+1} is even, m_j is odd, hence the 2 additive term is present for S_j , and thus $\gamma_j \geq 1$. We can thus charge z_{j+1} to $w_{g(j+1)} + 1$ and to an uncharged vertex in $\mathcal{I}_B^j$, and all the required properties remain satisfied. In particular, z_1 was already charged to two vertices in $\mathcal{I}_B^j$, given that $\gamma_j \geq 1$. Also, $\gamma_{j+1} = \gamma_j - 1$, given that $\mathcal{I}_B^{j+1}$ contains 1 vertex more than $\mathcal{I}_B^j$ and that z_{j+1} is charged to two vertices in $\mathcal{I}_B^{j+1}$. Also note that $S_{j+1} = S_j - 1$, given that $\lfloor m_{j+1}/2 \rfloor = \lfloor m_j/2 \rfloor + 1$ and the 2 additive term is present for S_j , given that m_j is odd, but not for S_{j+1} , given that m_{j+1} is even. Since $\gamma_j = S_j$, by hypothesis, we have that $\gamma_{j+1} = S_{j+1}$.

This completes the proof that the described charging scheme satisfies the required properties. It remains to charge z_k , though. In order to do that, it suffices to prove that at least one of the following three holds true: (a) $\gamma_{k-1} \geq 2$, (b) $\gamma_{k-1} = 1$ and $d(z_k) \geq 3$, or (c) $d(z_k) \geq 4$. Indeed, if (a) holds then z_k can be charged to two uncharged vertices in $\mathcal{I}_B^{k-1}$, if (b) holds then z_k can be charged to an uncharged vertex in $\mathcal{I}_B^{k-1}$ and to w_s , and if (c) holds then z_k can be charged to w_{s-1} and to w_s . We distinguish three cases.

- Suppose first that $d(z_k) = 2$. If $d(z_1) \geq 5$, then $\gamma_{k-1} \geq 2$, as all the other additive terms in the definition of S_{k-1} are non-negative, and thus (a) holds. We can hence assume that $d(z_1) \leq 4$. We distinguish three further cases. In each case, we prove that $\gamma_{k-1} \geq 2$, which implies that (a) holds.
 - Suppose that $d(z_1) = 4$, hence γ_{k-1} gets a contribution of 1 from the first additive term in the definition of S_{k-1} . If there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) \geq 4$, then γ_{k-1} gets a contribution of at least 1 from the second additive term in the definition of S_{k-1} ; since the third and fourth terms are non-negative, we have $\gamma_{k-1} \geq 2$. We can thus assume that $d(z_i) \leq 3$, for $i = 2, \dots, k-1$. We cannot have $d(z_2) = d(z_3) = \dots = d(z_{k-1}) = 3$, as otherwise we would be Case 5.2.1(i), thus we have $m \geq 1$. If m is odd, then γ_{k-1} gets a

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

contribution of 2 from the fourth additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 3$. If $m \geq 2$, then γ_{k-1} gets a contribution of at least 1 from the third additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$.

- Suppose next that $d(z_1) = 3$, hence the first additive term in the definition of S_{k-1} is equal to 0. If there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) \geq 5$, then γ_{k-1} gets a contribution of at least 2 from the second additive term in the definition of S_{k-1} ; since the third and fourth terms are non-negative, we have $\gamma_{k-1} \geq 2$. We can thus assume that $d(z_i) \leq 4$, for $i = 2, \dots, k-1$. If we have two distinct vertices z_i and $z_{i'}$ with $i, i' \in \{2, \dots, k-1\}$ and $d(z_i) = 4$ and $d(z_{i'}) = 4$, then again γ_{k-1} gets a contribution of at least 2 from the second additive term in the definition of S_{k-1} and we have $\gamma_{k-1} \geq 2$. We cannot have $d(z_2) = d(z_3) = \dots = d(z_{k-1}) = 3$, as otherwise we would be in Case 5.2.1(i), and we cannot have $d(z_i) = 4$ and $d(z_j) = 3$, for some $i \in \{2, \dots, k-1\}$ and for $j = 2, \dots, k-1$ with $j \neq i$, as otherwise we would be in Case 5.2.1(ii), thus we have $m \geq 1$. If m is odd, then γ_{k-1} gets a contribution of 2 from the fourth additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$. If $m \geq 4$, then γ_{k-1} gets a contribution of at least 2 from the third additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$. We are left with the case $m = 2$, in which γ_{k-1} gets a contribution of 1 from the third additive term in the definition of S_{k-1} and of 0 from the fourth term. We cannot have $d(z_j) \leq 3$ for $j = 2, \dots, k-1$, as otherwise we would be in Case 5.2.1(v), thus we have $d(z_i) = 4$, for some $i \in \{2, \dots, k-1\}$, hence γ_{k-1} gets a contribution of 1 from the second additive term in the definition of S_{k-1} , thus $\gamma_{k-1} \geq 2$.
- Suppose next that $d(z_1) = 2$, hence the first additive term in the definition of S_{k-1} is equal to -1 .
 - * If there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) \geq 6$, then γ_{k-1} gets a contribution of at least 3 from the second additive term in the definition of S_{k-1} ; since the third and fourth terms are non-negative, we have $\gamma_{k-1} \geq 2$. We can thus assume that $d(z_i) \leq 5$, for $i = 2, \dots, k-1$.
 - * Suppose that there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) = 5$, hence γ_{k-1} gets a contribution of at least 2 from the second additive term in the definition of S_{k-1} . If $m \geq 1$, then γ_{k-1} gets a contribution of at least 1 from the third or

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

fourth additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$. We cannot have $d(z_j) = 3$, for $j = 2, \dots, k-1$ with $j \neq i$, as otherwise we would be in Case 5.2.1(iii). If there exists a vertex $z_{i'}$ with $i' \in \{2, \dots, k-1\}$, $i' \neq i$, and $d(z_{i'}) \geq 4$, then γ_{k-1} gets a contribution of at least 3 from the second additive term in the definition of S_{k-1} , thus we have $\gamma_{k-1} \geq 2$. We can thus assume that $d(z_i) \leq 4$, for $i = 2, \dots, k-1$.

- * Suppose that there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) = 4$, hence γ_{k-1} gets a contribution of at least 1 from the second additive term in the definition of S_{k-1} . If m is odd, then γ_{k-1} gets a contribution of 2 from the fourth additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$. If $m \geq 4$, then γ_{k-1} gets a contribution of at least 2 from the third additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$. We cannot have $m = 2$ and $d(z_j) \leq 3$, for $j = 2, \dots, k-1$ with $j \neq i$, as otherwise we would be in Case 5.2.1(vii). If $m = 2$ and $d(z_{i'}) = 4$, for some $i' \in \{2, \dots, k-1\}$ with $i' \neq i$, then γ_{k-1} gets a contribution of at least 2 from the second additive term in the definition of S_{k-1} and a contribution of 1 from the third additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$. This leaves us with the case $m = 0$. We cannot have $d(z_j) = 3$, for $j = 2, \dots, k-1$ with $j \neq i$, as otherwise we would be in Case 5.2.1(ii). Also, we cannot have $d(z_{i'}) = 4$ and $d(z_j) = 3$, for some $i' \in \{2, \dots, k-1\}$ with $i' \neq i$ and for $j = 2, \dots, k-1$ with $j \neq i$ and $j \neq i'$, as otherwise we would be in Case 5.2.1(iv). It follows that we have $d(z_{i'}) = 4$ and $d(z_{i''}) = 4$, for some $i', i'' \in \{2, \dots, k-1\}$ with $i' \neq i$, $i'' \neq i$ and $i'' \neq i'$. Hence, γ_{k-1} gets a contribution of at least 3 from the second additive term in the definition of S_{k-1} , thus we have $\gamma_{k-1} \geq 2$. We can thus assume that $d(z_i) \leq 3$, for $i = 2, \dots, k-1$.
- * We cannot have $m = 0$, as otherwise we would be in Case 5.1 (observe that we would have $k \geq 3$, since G is internally-triangulated). Also, we cannot have $m = 2$ or $m = 4$, as otherwise we would be in Case 5.2.1(v). Furthermore, we cannot have $m = 1$. Indeed, if $k = 3$, we would be in Case 5.1, if $k \geq 4$ and $d(z_{k-1}) = 3$ we would be in Case 5.2.1(vi), and if $k \geq 4$ and $d(z_2) = 3$ we would be in Case 5.2.2(vi); note that $d(z_{k-1}) = 2$ and $d(z_2) = 2$ cannot hold simultaneously,

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

given that $m = 1$. If $m \geq 3$ and m is odd, then γ_{k-1} gets a contribution of 2 from the fourth additive term in the definition of S_{k-1} , and of at least 1 from the third additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$. Finally, if $m \geq 6$, then γ_{k-1} gets a contribution of at least 3 from the third additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 1$.

- Suppose next that $d(z_k) = 3$. If $d(z_1) \geq 4$, then $\gamma_{k-1} \geq 1$, as all the other additive terms in the definition of S_{k-1} are non-negative, and thus (b) holds. We can hence assume that $d(z_1) \leq 3$. We distinguish two further cases. In each case, we prove that $\gamma_{k-1} \geq 1$, which implies that (b) holds.
 - Suppose first that $d(z_1) = 3$, hence the first additive term in the definition of S_{k-1} is equal to 0. If there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) \geq 4$, then γ_{k-1} gets a contribution of at least 1 from the second additive term in the definition of S_{k-1} ; since the third and fourth terms are non-negative, we have $\gamma_{k-1} \geq 1$. We can thus assume that $d(z_i) \leq 3$, for $i = 2, \dots, k-1$. We cannot have $d(z_2) = d(z_3) = \dots = d(z_{k-1}) = 3$, as otherwise B would be a pesky component, thus we have $d(z_i) = 2$, for some $i \in \{2, \dots, k-1\}$. However, this implies that γ_{k-1} gets a contribution of at least 1 from the third or fourth additive term in the definition of S_{k-1} , thus we have $\gamma_{k-1} \geq 1$.
 - Suppose finally that $d(z_1) = 2$, hence the first additive term in the definition of S_{k-1} is equal to -1 .
 - * If there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) \geq 5$, then γ_{k-1} gets a contribution of at least 2 from the second additive term in the definition of S_{k-1} ; since the third and fourth terms are non-negative, we have $\gamma_{k-1} \geq 1$. We can thus assume that $d(z_i) \leq 4$, for $i = 2, \dots, k-1$.
 - * Suppose that there exists a vertex z_i with $i \in \{2, \dots, k-1\}$ and $d(z_i) = 4$, hence γ_{k-1} gets a contribution of at least 1 from the second additive term in the definition of S_{k-1} . If $m \geq 1$, then γ_{k-1} gets a contribution of at least 1 from the third or fourth additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 1$. We cannot have $d(z_j) = 3$, for $j = 2, \dots, k-1$ with $j \neq i$, as otherwise we would be in Case 5.2.2(ii). If there exists a vertex $z_{i'}$ with $i' \in \{2, \dots, k-1\}$, $i' \neq i$, and $d(z_{i'}) \geq 4$, then γ_{k-1} gets a contribution of at least 2 from the second additive term in the

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

definition of S_{k-1} , thus we have $\gamma_{k-1} \geq 1$. We can thus assume that $d(z_i) \leq 3$, for $i = 2, \dots, k-1$.

- * We cannot have $m = 0$, as otherwise we would be in Case 5.2.2(i). Also, we cannot have $m = 2$, as otherwise we would be in Case 5.2.2(v). If m is odd, then γ_{k-1} gets a contribution of 2 from the fourth additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 1$. Finally, if $m \geq 4$, then γ_{k-1} gets a contribution of at least 2 from the third additive term in the definition of S_{k-1} , hence $\gamma_{k-1} \geq 2$.

- Suppose finally that $d(z_k) \geq 4$. Then (c) holds.

This concludes the proof that $\mathcal{I}_B$ satisfies Property A and thus the proof of the lemma. $\square$

We can now apply induction as follows. Let $\mathcal{I}_B$ be a set as in [Lemma 8.10](#). If $\mathcal{I}_B$ satisfies Property A, let H be the subgraph of G induced by $V(G) - (V(G_B) - \{c_B\})$; otherwise, if it satisfies Property B, let H be the subgraph of G induced by $V(G) - (V(G_B) - \{c_B, \ell_B\})$; otherwise, it satisfies Property C and then let H be the subgraph of G induced by $V(G) - (V(G_B) - \{c_B, r_B\})$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \mathcal{I}_B$. We prove that $\mathcal{I}$ is a good set. Note that $|\mathcal{I}| = |\mathcal{I}_H| + |\mathcal{I}_B|$. If $\mathcal{I}_B$ satisfies Property A, then $|V(H)| = n - n_B + 1$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - n_B + 1)/3$ and, by [Lemma 8.10](#), we have $|\mathcal{I}_B| \geq 2(n_B - 1)/3$, thus $|\mathcal{I}| \geq 2n/3$. If $\mathcal{I}_B$ satisfies Property B or Property C, then $|V(H)| = n - n_B + 2$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - n_B + 2)/3$ and, by [Lemma 8.10](#), we have $|\mathcal{I}_B| \geq 2(n_B - 2)/3$, thus $|\mathcal{I}| \geq 2n/3$. In order to prove that $G[\mathcal{I}]$ is outerplane, note that $G[\mathcal{I}_H]$ is outerplane by induction. If $\mathcal{I}_B$ satisfies Property A, then $G[\mathcal{I}_B \cup (\{c_B\} \cap \mathcal{I}_H)]$ is outerplane by [Lemma 8.10](#). Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_B \cup (\{c_B\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one and share either no vertex or one vertex, by [Observation 8.2](#) we have that $G[\mathcal{I}]$ is an outerplane graph. If $\mathcal{I}_B$ satisfies Property B, then $G[\mathcal{I}_B \cup (\{c_B, \ell_B\} \cap \mathcal{I}_H)]$ is outerplane by [Lemma 8.10](#). Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_B \cup (\{c_B, \ell_B\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#) we have that $G[\mathcal{I}]$ is an outerplane graph. Similarly, if $\mathcal{I}_B$ satisfies Property C, then $G[\mathcal{I}_B \cup (\{c_B, r_B\} \cap \mathcal{I}_H)]$ is outerplane by [Lemma 8.10](#). Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_B \cup (\{c_B, r_B\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#) we have that $G[\mathcal{I}]$ is an outerplane graph.

As long as B is a non-trivial block, we can now assume that B is pesky.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

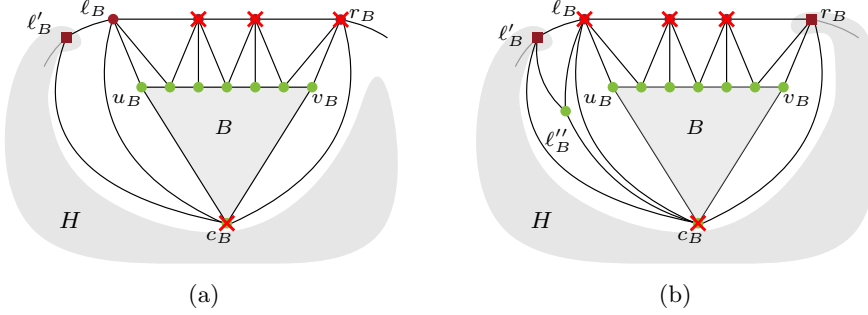


Figure 8.18: Illustrations for (a) Case 6 and (b) Case 7.

Case 6: B is non-trivial and pesky, the edge $c_B \ell'_B$ exists, and the cycle $c_B \ell_B \ell'_B$ bounds an internal face of G . Refer to Figure 8.18a. Let H be the subgraph of G induced by $V(G) - V(G_B)$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . Let $\mathcal{I}_B = \{\ell_B\} \cup (V(B) - \{c_B\})$. We define $\mathcal{I} = \mathcal{I}_H \cup \mathcal{I}_B$. Note that c_B and r_B do not belong to $\mathcal{I}$, that ℓ_B belongs to $\mathcal{I}$, and that ℓ'_B might or might not belong to $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(B)|$. By induction, $|\mathcal{I}_H| \geq 2|V(H)|/3$; also, by Observation 8.5, $|V(B)| = 2|V(G_B)|/3$. Hence, $|\mathcal{I}| \geq 2(|V(H)| + |V(G_B)|)/3 = 2n/3$. Also, $G[\mathcal{I}_H]$ is outerplane by induction and $G[\mathcal{I}_B \cup (\{\ell'_B\} \cap \mathcal{I}_H)]$ is outerplane, as well, given that B is an outerplane graph, given that ℓ_B is outside B , where $\ell_B u_B u'_B$ bounds a face of G and ℓ_B is not adjacent to any vertex of $\mathcal{I}_B$ different from u_B and u'_B , and given that ℓ'_B is outside $G[\mathcal{I}_B]$ and not adjacent to any vertex of $\mathcal{I}_B$ different from ℓ_B . Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_B \cup (\{\ell'_B\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one, and share either no vertex or a single vertex, by Observation 8.2, we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 6': B is non-trivial and pesky, the edge $c_B r'_B$ exists, and the cycle $c_B r_B r'_B$ bounds an internal face of G . This case can be discussed symmetrically to Case 6.

Case 7: B is non-trivial and pesky, the edge $c_B \ell'_B$ exists, and the cycle $c_B \ell_B \ell'_B$ contains in its interior a unique vertex. Refer to Figure 8.18b. Let ℓ''_B be the vertex contained in the cycle $c_B \ell_B \ell'_B$. Let H be the subgraph of G induced by $V(G) - (V(G_B) - \{r_B\} \cup \{\ell''_B\})$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . Let $\mathcal{I}_B = \{\ell''_B\} \cup (V(B) - \{c_B\})$. We define $\mathcal{I} = \mathcal{I}_H \cup \mathcal{I}_B$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

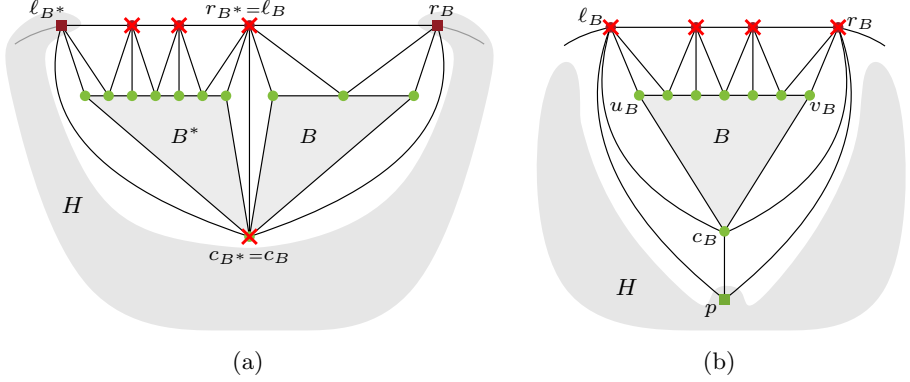


Figure 8.19: Illustrations for (a) Case 8 and (b) Case 11.

Note that c_B and ℓ_B do not belong to $\mathcal{I}$, while each of ℓ'_B and r_B might or might not belong to $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(B)|$, hence $|\mathcal{I}| \geq 2n/3$ follows as in Case 6. Also, $G[\mathcal{I}_H]$ is outerplane by induction and $G[V(B) - \{c_B\} \cup (\{r_B\} \cap \mathcal{I}_H)]$ is outerplane, as well, given that B is an outerplane graph and given that r_B is outside B , where $r_B v_B v'_B$ bounds a face of G and r_B is not adjacent to any vertex of $V(B) - \{c_B\}$ different from v_B and v'_B . Since $G[\mathcal{I}_H]$ and $G[V(B) - \{c_B\} \cup (\{r_B\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one, and share either no vertex or a single vertex, by [Observation 8.2](#), we have that $G[\mathcal{I} - \{\ell'_B\}]$ is an outerplane graph. Furthermore, ℓ'_B and ℓ''_B induce an edge, hence an outerplane graph. Since $G[\mathcal{I} - \{\ell''_B\}]$ and $G[\{\ell''_B\} \cup (\mathcal{I}_H \cap \{\ell'_B\})]$ are each in the outer face of the other one and share either no vertex or a single vertex, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 7': B is non-trivial and pesky, the edge $c_B r'_B$ exists, and the cycle $c_B r_B r'_B$ contains in its interior a unique vertex r''_B . This case can be discussed symmetrically to Case 7.

Case 8: B is non-trivial and pesky, and there exists a non-trivial pesky extremal leaf $B^* \neq B$ in K such that the link-vertex c_{B^*} of B^* is c_B and the right cage vertex r_{B^*} of B^* is ℓ_B . Refer to [Figure 8.19a](#). Let H be the subgraph of G induced by $V(G) - (V(G_B) - \{r_B\}) - (V(G_{B^*}) - \{\ell_{B^*}\})$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . Let $\mathcal{I}_B = (V(B) \cup V(B^*)) - \{c_B\}$. We define $\mathcal{I} = \mathcal{I}_H \cup \mathcal{I}_B$. Note that $c_B = c_{B^*}$ and $r_{B^*} = \ell_B$ do not belong to $\mathcal{I}$,

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

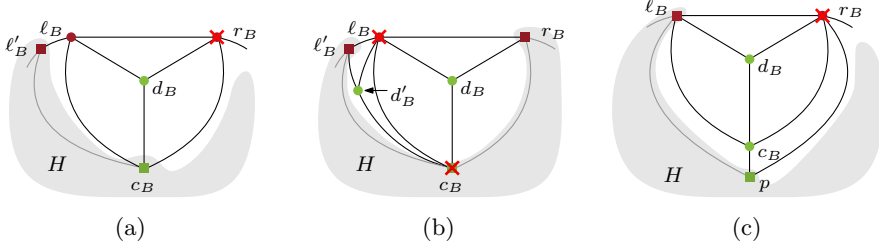


Figure 8.20: Illustrations for (a) Case 9, (b) Case 10, and (c) Case 12.

while each of ℓ_{B^*} and r_B might or might not belong to $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(B)|$. By induction, $|\mathcal{I}_H| \geq 2|V(H)|/3$; also, by [Observation 8.5](#), $|V(B)| = 2|V(G_B)|/3$ and $|V(B^*)| = 2|V(G_{B^*})|/3$. The number of vertices of H is $n - (|V(G_B)| + |V(G_{B^*})| - 4)$, since the vertices of G that do not belong to H are those of G_B and G_{B^*} , except for $\ell_{B^*} = \ell_B$ and $r_B = c_B$. Also note that $|\mathcal{I}_B| = |V(B)| + |V(B^*)| - 2$, as c_B belongs both to B and B^* and is not part of $\mathcal{I}_B$. It follows that $|\mathcal{I}| \geq 2(n - |V(G_B)| - |V(G_{B^*})| + 4)/3 + |V(B)| + |V(B^*)| - 2 = 2(n - |V(G_B)| - |V(G_{B^*})|)/3 + 2(|V(G_B)| + |V(G_{B^*})|)/3 + 8/3 - 2 = 2n/3 + 2/3 > 2n/3$. Also, $G[\mathcal{I}_H]$ is outerplane by induction and $G[(V(B) - \{c_B\}) \cup (\{r_B\} \cap \mathcal{I}_H)]$ is outerplane, as well, given that r_B is outside B , that $r_B v_B v'_B$ bounds a face of G , and that r_B is not adjacent to any vertex of $V(B) - \{c_B\}$ different from v_B and v'_B . Since $G[\mathcal{I}_H]$ and $G[(V(B) - \{c_B\}) \cup (\{r_B\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one, and share either no vertex or a single vertex, by [Observation 8.2](#), we have that $G[\mathcal{I}_H \cup (V(B) - \{c_B\})]$ is an outerplane graph. Similarly, $G[(V(B^*) - \{c_B\}) \cup (\{\ell_{B^*}\} \cap \mathcal{I}_H)]$ is outerplane, and since $G[\mathcal{I}_H \cup (V(B) - \{c_B\})]$ and $G[(V(B^*) - \{c_B\}) \cup (\{\ell_{B^*}\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one, and share either no vertex or a single vertex, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 8': B is non-trivial and pesky, and there exists a non-trivial pesky extremal leaf $B^* \neq B$ in K such that the link-vertex c_{B^*} of B^* is c_B and the left cage vertex ℓ_{B^*} of B^* is r_B . This case can be discussed symmetrically to Case 8.

Suppose next that B is a trivial block; recall that, in this case, B is an edge $c_B d_B$, where c_B is a cutvertex of K .

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

Case 9: B is trivial, the edge $c_B\ell'_B$ exists, and the cycle $c_B\ell_B\ell'_B$ bounds an internal face of G . Refer to [Figure 8.20a](#). Let H be the subgraph of G induced by $V(G) - \{\ell_B, r_B, d_B\}$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \{\ell_B, d_B\}$. Note that r_B does not belong to $\mathcal{I}$, while c_B and ℓ'_B might or might not belong to $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + 2$. Note that $|V(H)| = n - 3$, hence, by induction, we have $|\mathcal{I}_H| \geq 2(n - 3)/3$. It follows that $|\mathcal{I}| \geq 2(n - 3)/3 + 2 = 2n/3$. Also, $G[\mathcal{I}_H]$ is outerplane by induction and the subgraph of G induced by c_B, ℓ'_B, ℓ_B , and d_B is outerplane, as well, since it is composed of the cycles $c_B\ell_B\ell'_B$ and $c_Bd_B\ell_B$, with d_B outside the former cycle. Since $G[\mathcal{I}_H]$ and $G[\{c_B, \ell'_B, \ell_B, d_B\} \cap \mathcal{I}_H]$ are each in the outer face of the other one, and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 9': B is trivial, the edge $c_Br'_B$ exists and the cycle $c_Br_Br'_B$ bounds an internal face of G . This case can be discussed symmetrically to Case 9.

Case 10: B is trivial, the edge $c_B\ell'_B$ exists and the cycle $c_B\ell_B\ell'_B$ contains in its interior a unique vertex. Refer to [Figure 8.20b](#). Let d'_B be the vertex contained in the cycle $c_B\ell_B\ell'_B$. Let H be the subgraph of G induced by $V(G) - \{\ell_B, d_B, d'_B\}$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \{d_B, d'_B\}$. Note that ℓ_B does not belong to $\mathcal{I}$, while c_B, r_B , and ℓ'_B might or might not belong to $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + 2$ and $|V(H)| = n - 3$, hence $|\mathcal{I}| \geq 2n/3$ follows as in Case 6. Also, $G[\mathcal{I}_H]$ is outerplane by induction, and $G[\{c_B, \ell'_B, d'_B\}]$ and $G[\{c_B, r_B, d_B\}]$ are outerplane graphs, as well, since they are cycles. Since $G[\mathcal{I}_H]$ and $G[\{c_B, \ell'_B, d'_B\} \cap \mathcal{I}_H]$ are each in the outer face of the other one, and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#), we have that $G[\mathcal{I}_H \cup \{d'_B\}]$ is an outerplane graph. Also, since $G[\mathcal{I}_H \cup \{d'_B\}]$ and $G[\{c_B, r_B, d_B\} \cap \mathcal{I}_H]$ are each in the outer face of the other one, and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 10': B is trivial, the edge $c_Br'_B$ exists, and the cycle $c_Br_Br'_B$ contains in its interior a unique vertex. This case can be discussed symmetrically to Case 10.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

By [Lemma 8.6](#), we can now assume that B is the only child of c_B in T_K . The next two cases deal with the situation in which the parent B_P of c_B in T_K is a trivial block and B is non-trivial or trivial, respectively.

Case 11: B is non-trivial and pesky, and the parent B_P of c_B in T_K is a trivial block. Refer to [Figure 8.19b](#). Let pc_B be the edge corresponding to B_P . By [Lemma 8.7](#), the edges $p\ell_B$ and pr_B belong to G and the cycles $p\ell_B c_B$ and $pr_B c_B$ delimit faces of G . Let H be the subgraph of G induced by $V(G) - V(G_B)$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . Let $\mathcal{I}_B = V(B)$. We define $\mathcal{I} = \mathcal{I}_H \cup \mathcal{I}_B$. Note that ℓ_B and r_B do not belong to $\mathcal{I}$, that c_B belongs to $\mathcal{I}$, and that p might or might not belong to $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + |V(B)|$, hence $|\mathcal{I}| \geq 2n/3$ follows as in Case 6. Also, $G[\mathcal{I}_H]$ is outerplane by induction and $G[\mathcal{I}_B \cup (\{p\} \cap \mathcal{I}_H)]$ is outerplane, as well, given that B is an outerplane graph and given that the edge pc_B is outside B . Since $G[\mathcal{I}_H]$ and $G[\mathcal{I}_B \cup (\{p\} \cap \mathcal{I}_H)]$ are each in the outer face of the other one, and share either no vertex or a single vertex, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 12: B is trivial and the parent B_P of c_B in T_K is a trivial block. Refer to [Figure 8.20c](#). Let pc_B be the edge corresponding to the parent of c_B in T_K . By [Lemma 8.7](#), the edges $p\ell_B$ and pr_B belong to G and the cycles $p\ell_B c_B$ and $pr_B c_B$ delimit faces of G . Let H be the subgraph of G induced by $V(G) - \{c_B, d_B, r_B\}$. We apply induction on H , so to find a good set $\mathcal{I}_H$ for H . We define $\mathcal{I} = \mathcal{I}_H \cup \{c_B, d_B\}$. Note that r_B does not belong to $\mathcal{I}$, while p and ℓ_B might or might not belong to $\mathcal{I}$. We prove that $\mathcal{I}$ is a good set. First, we have $|\mathcal{I}| = |\mathcal{I}_H| + 2$ and $|V(H)| = n - 3$, hence $|\mathcal{I}| \geq 2n/3$ follows as in Case 6. Also, $G[\mathcal{I}_H]$ is outerplane by induction and $G[\{p, c_B, d_B, \ell_B\}]$ is outerplane, as well, since it is composed of the cycles $pc_B \ell_B$ and $c_B d_B \ell_B$, with d_B outside the former cycle. Since $G[\mathcal{I}_H]$ and $G[\{p, c_B, \ell_B, d_B\} \cap \mathcal{I}_H]$ are each in the outer face of the other one, and share either no vertex, or a single vertex, or a single edge, by [Observation 8.2](#), we have that $G[\mathcal{I}]$ is an outerplane graph.

Case 13: Neither of Cases 1–12 applies. Let K be a terminal component of $G[L_2]$ and B_P be a biconnected component of K as in [Lemma 8.8](#). If B_P is not the root of T_K , then let c_P be the parent of B_P in T_K . Otherwise, let l be the leaf of T^* corresponding to the face of $G[L_1]$ in which K lies, and let xy be the edge of $G[L_1]$ dual to the edge of T^* from l to its parent (or let $e^* = xy$

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

if l is the root of T^*). Then c_P is the vertex of K such that the 3-cycle $xy c_P$ bounds an internal face of G .

Let u_P and v_P be the neighbors of c_P on the boundary of the outer face of B_P , where u_P , c_P , and v_P appear in this counter-clockwise order along the outer face of B_P . Let $c_P \ell_P$ be the edge of G that precedes $c_P u_P$ in clockwise direction around c_P and note that $\ell_P \in L_1$. Symmetrically, let $c_P r_P$ be the edge of G that follows $c_P v_P$ in clockwise direction around c_P and note that $r_P \in L_1$. Note that, since G is internally-triangulated, the edges $\ell_P u_P$ and $r_P v_P$ exist and the cycles $c_P \ell_P u_P$ and $c_P r_P v_P$ bound internal faces of G .

Let G_P be the subgraph of G whose vertices and edges are inside or on the boundary of the cycle composed of the path $\ell_P c_P r_P$ and of the path obtained by traversing in clockwise direction the outer face of G from ℓ_P to r_P . Let $n_P = |V(G_P)|$. Although the setting is different, the general strategy we employ to handle this final case is similar to the one of Cases 4 and 5.2 and supported by the following lemma.

Lemma 8.11. *There exists a set $\mathcal{I}_P \subset V(G_P)$ such that $\{c_P, \ell_P, r_P\} \cap \mathcal{I}_P = \emptyset$ and such that (at least) one of the following properties is satisfied:*

- Property A: $|\mathcal{I}_P| \geq 2(n_P - 1)/3$ and $G[\mathcal{I}_P \cup \{c_P\}]$ is an outerplane graph;
- Property B: $|\mathcal{I}_P| \geq 2(n_P - 2)/3$ and $G[\mathcal{I}_P \cup \{c_P, \ell_P\}]$ is an outerplane graph; and
- Property C: $|\mathcal{I}_P| \geq 2(n_P - 2)/3$ and $G[\mathcal{I}_P \cup \{c_P, r_P\}]$ is an outerplane graph.

Proof. Refer to [Figure 8.21](#). Let $c_P, w_1 = u_P, w_2, \dots, w_{s-1}, w_s = v_P$ be the clockwise order of the vertices along the outer face of B_P , and let $c_P, z_1 = \ell_P, z_2, \dots, z_{k-1}, z_k = r_P$ be the clockwise order of the vertices along the outer face of G_P . Let K_P be the subgraph of K composed of B_P and of all the blocks that are descendants of B_P ; that is, K_P is obtained from G_P by removing the vertices $z_1, z_2, \dots, z_k$ and their incident edges. Note that every vertex among $z_1, z_2, \dots, z_k$ has at least two neighbors in K_P . Indeed, z_1 is adjacent to c_P and u_P , z_k is adjacent to c_P and v_P , and, for any $i \in \{2, \dots, k-1\}$, vertex z_i has at least two neighbors in K_P , as otherwise Case 1 or Case 2 would apply.

We say that a vertex z_j is *transparent* if either: (i) z_j has exactly two neighbors in K_P ; or (ii) z_j has exactly three neighbors in K_P , one of which is a vertex d_B that has degree 1 in K_P . Note that, if (i) holds true, as for z_1

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

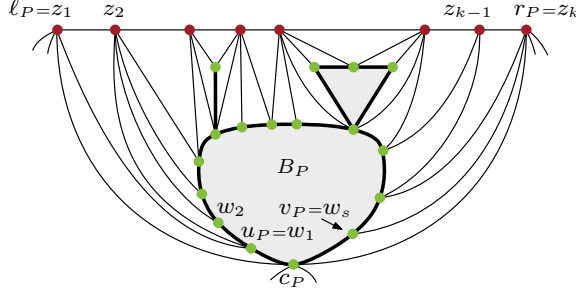


Figure 8.21: Illustration for Case 13. The figure depicts G_P , plus part of edges incident to ℓ_P , c_P , and r_P outside G_P . The boundary of K_P is represented by thick lines.

and z_{k-1} in Figure 8.21, then the two neighbors of z_j in K_P belong to B_P . Indeed, every non-trivial block B that is a descendant of B_P in T_K is pesky, hence if z_j has a neighbor in B it actually has three neighbors in B . Also, if z_j has a neighbor d_B that does not belong to B_P and belongs to a trivial block B , then $z_j = \ell_B$ or $z_j = r_B$, hence z_j is also a neighbor of the parent c_B of B in T_K and of c_B^l or c_B^r , respectively, by Lemma 8.8. If (ii) holds true, as for z_3 in Figure 8.21, then d_B is a vertex of a trivial block B that is child of a child c_B of B_P , with $d_B \neq c_B$, and z_j is neighbor of d_B , c_B , and of either c_B^l or c_B^r . We say that z_j is *opaque* if it is not transparent.

Observe that $k \geq 2$. Namely, by Lemma 8.8, we have that B_P has at least one child c_B in T_K , and c_B has one child B ; then, if $k = 1$, the edges $c_B \ell_B$ and $c_B r_B$, which exist since G is internally-triangulated, would connect the same pair of vertices. We start by considering the case $k = 2$. In this case, we can define $\mathcal{I}_P = V(K_P) - \{c_P\}$ and observe that $\{c_P, \ell_P, r_P\} \cap \mathcal{I}_P = \emptyset$. If $|V(K_P)| \geq 5$ (see Figure 8.22a), then $\mathcal{I}_P$ satisfies Property A. Namely, $G[\mathcal{I}_P \cup \{c_P\}]$ is an outerplane graph, because it coincides with K_P . Since $n_P = |V(K_P)| + 2 \geq 7$, it follows that $|\mathcal{I}_P| = |V(K_P)| - 1 = n_P - 3 \geq 2(n_P - 1)/3$, as required. If $|V(K_P)| \leq 4$ (see Figure 8.22b), then, since B_P is a non-trivial block that is not a leaf in T_K , we have that $|V(K_P)| = 4$, that $|V(B_P)| = 3$, and that B_P has exactly one descendant block B in T_K , where B is a trivial block. Suppose that the cutvertex parent of B is u_P , as the case in which it is v_P is symmetric. By Lemma 8.8, we have that G_P contains the edges $\ell_P u_P$, $r_P u_P$, and $r_P v_P$. Then $\mathcal{I}_P$ satisfies Property B (if the cutvertex parent of B is v_P , then $\mathcal{I}_P$ satisfies Property C). Namely, we have that $G[\mathcal{I}_P \cup \{c_P, \ell_P\}]$ is an outerplane graph,

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

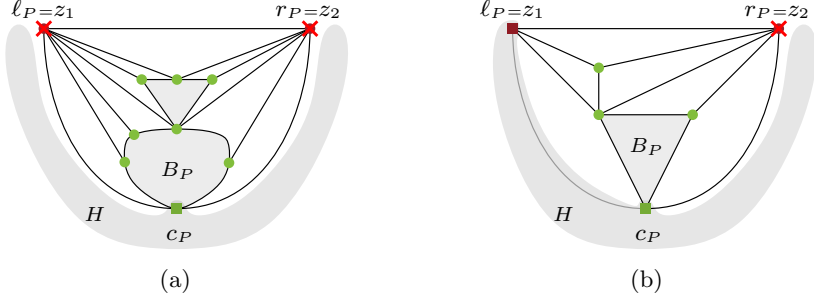


Figure 8.22: Illustrations for Case 13 when $k = 2$ (a) if $|V(K_P)| \geq 5$ and (b) if $|V(K_P)| \leq 4$.

since all its vertices are adjacent to r_P , which is in L_1 and not in $\mathcal{I}_P \cup \{c_P, \ell_P\}$. Also, $|\mathcal{I}_P| = 3 > 2(n_P - 2)/3 = 8/3$, as required.

We can now assume that $k \geq 3$. In this case, the construction of $\mathcal{I}_P$ is similar to Cases 4 and 5.2. First, all the vertices of B_P , except for c_P , are in $\mathcal{I}_P$. Second, ℓ_P and r_P are not in $\mathcal{I}_P$, hence $\{c_P, \ell_P, r_P\} \cap \mathcal{I}_P = \emptyset$. Third, each opaque vertex z_j is not in $\mathcal{I}_P$. Finally, whether a transparent vertex is in $\mathcal{I}_P$ or not is decided according to five different cases.

In **Case 13.1**, z_1 is transparent and z_2 is opaque. Then, for $j = 3, \dots, k-1$, if z_j is a transparent vertex, it is in $\mathcal{I}_P$ if $z_{j-1} \notin \mathcal{I}_P$, and it is not in $\mathcal{I}_P$ if $z_{j-1} \in \mathcal{I}_P$.

We prove that $\mathcal{I}_P$ satisfies Property B. In order to prove that $G[\mathcal{I}_P \cup \{c_P, \ell_P\}]$ is an outerplane graph, it suffices to show that, for each vertex in $V(K_P) - \{c_P\}$, there exists a neighbor among $z_1, \dots, z_k$ which is not in $\mathcal{I}_P \cup \{c_P, \ell_P\}$. This is done similarly to the proofs of [Lemmas 8.9](#) and [8.10](#). Namely, consider a vertex w in $V(K_P) - \{c_P\}$. If w is a neighbor of an opaque vertex z_j , then z_j is not in $\mathcal{I}_P \cup \{c_P, \ell_P\}$, hence it is the desired neighbor of w ; this includes all vertices w that belong to non-trivial (pesky) blocks which are descendants of B_P in T_K . Each remaining vertex w is only neighbor of transparent vertices. Note that, for any two transparent vertices z_j and z_{j+1} that are consecutive in the sequence $z_1, \dots, z_k$, we have that at most one of them belongs to $\mathcal{I}_P \cup \{c_P, \ell_P\}$. This comes from the algorithm's construction if $3 \leq j \leq k-2$, from the fact that r_P is not in $\mathcal{I}_P \cup \{c_P, \ell_P\}$ if $j = k-1$ and from the assumption that z_2 is opaque if $j = 1$ or $j = 2$. If w belongs to a trivial block B , then its neighbors ℓ_B and r_B are consecutive in the sequence $z_1, \dots, z_k$, hence one of them does not belong to $\mathcal{I}_P \cup \{c_P, \ell_P\}$. Also, if w is a vertex w_i of B_P and does not

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

belong to any block that is a descendant of B_P in T_K , then again its neighbors are consecutive in the sequence $z_1, \dots, z_k$, hence one of them does not belong to $\mathcal{I}_P \cup \{c_P, \ell_P\}$.

We prove that $|\mathcal{I}_P| \geq 2(n_P - 2)/3$. We charge each vertex of G_P not in $\mathcal{I}_P$ and different from c_P and ℓ_P to two vertices in $\mathcal{I}_P$, so that each vertex in $\mathcal{I}_P$ is charged with at most one vertex. The charging scheme is defined iteratively on the vertices $z_2, \dots, z_k$ that do not belong to $\mathcal{I}_P$. Throughout the algorithm that defines the charging scheme, we maintain an *active edge* $w_i z_j$, with the following meaning. Let $\mathcal{C}^{i,j}$ be the cycle $c_P w_1 w_2 \dots w_i z_j z_{j-1} \dots z_1$ and let $\mathcal{I}_P^{i,j}$ be the subset of $\mathcal{I}_P$ composed of those vertices that are inside or on the boundary of $\mathcal{C}^{i,j}$. Then the following properties are satisfied:

- (X1) All the vertices among $z_2, \dots, z_j$ that are not in $\mathcal{I}_P^{i,j}$ have been charged to two vertices in $\mathcal{I}_P^{i,j}$, so that each vertex in $\mathcal{I}_P^{i,j}$ has been charged with at most one vertex.
- (X2) If $j < k$, then the set $\mathcal{I}_P^{i,j}$ contains a *first free vertex* ϕ_1 , that is, a vertex that has not been charged with any vertex among $z_2, \dots, z_j$.
- (X3) If $j < k$ and if all the vertices of a block B that is a descendant of B_P are inside or on the boundary of $\mathcal{C}^{i,j}$, then $\mathcal{I}_P^{i,j}$ contains a *second free vertex* ϕ_2 , that is, a vertex that has not been charged with any vertex among $z_2, \dots, z_j$ and is different from ϕ_1 .
- (X4) We have that $j = 1$ or that z_j is not in $\mathcal{I}_P^{i,j}$.

Observe that, once $j = k$, Property (X1) implies that $|\mathcal{I}_P| \geq 2(n_P - 2)/3$.

The first active edge is $w_1 z_1$. Recall that the edge $w_1 z_1 = u_P \ell_P$ exists and that the cycle $c_P w_1 z_1$ bounds an internal face of G . Hence, Property (X3) is vacuously satisfied, since no block B that is a descendant of B_P has all its vertices inside or on the boundary of $\mathcal{C}^{1,1}$. Properties (X1) and (X4) are also trivially satisfied, since $j = 1$. Note that $z_1 = \ell_P$ does not need to be charged to any vertex. Finally, Property (X2) is satisfied by setting $\phi_1 = w_1$; observe that $\mathcal{I}_P^{1,1} = \{w_1\}$.

Suppose now that our charging scheme has defined an active edge $w_i z_j$, for some $1 \leq i \leq s$ and $1 \leq j \leq k - 1$, so that Properties (X1)–(X4) are satisfied. Let $w_i z_j v^*$ be the cycle delimiting an internal face of G that does not lie inside $\mathcal{C}^{i,j}$. We cannot have $v^* = c_P$, since this would imply that $i = s$ and $j = k$, given that the edge $c_P z_k$ follows $c_P w_s$ in clockwise order around c_P in G , by construction. We distinguish some cases.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

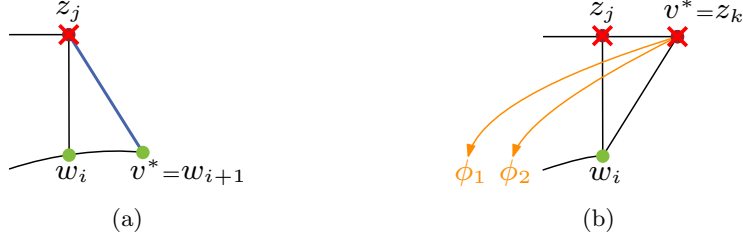


Figure 8.23: Illustrations for Case 13.1 (a) when $v^* = w_{i+1}$ and (b) when $v^* = z_k$. In this and the following figures, the thick blue edge denotes the new active edge.

- (I) First, if $v^* = w_{i+1}$ (see Figure 8.23a), we proceed with $w_{i+1}z_j$ as an active edge. Properties (X1)–(X4) are satisfied by $w_{i+1}z_j$ since they are satisfied by w_iz_j .
- (II) Second, suppose that $v^* = z_k$ (see Figure 8.23b). Then we conclude the algorithm by charging z_k to ϕ_1 and ϕ_2 . Observe that $\mathcal{C}^{i,j}$ contains inside or on its boundary all the vertices of a block B that is a descendant of B_P , hence $\mathcal{I}_P^{i,j}$ contains a second free vertex ϕ_2 . Indeed, by Lemma 8.8, we have that a block B that is a descendant of B_P exists. Furthermore, B cannot be contained inside the cycle $w_iw_{i+1}\dots w_sz_k$, as otherwise we would have $\ell_B = r_B = z_k$.
- (III) Third, suppose that $v^* = z_{j+1}$, with $j+1 < k$, and that $z_{j+1} \notin \mathcal{I}_P$ (see Figure 8.24). Then z_{j+1} is opaque. Indeed, we have that $j = 1$ or that z_j is not in $\mathcal{I}_P^{i,j}$. In the former case, z_{j+1} is opaque by assumption. In the latter case, if z_{j+1} were transparent, then it would belong to $\mathcal{I}_P$, by the algorithm's construction, whereas it does not. Note that there is no block B that is descendant of B_P , whose parent is w_i , and such that $\ell_B = z_{j+1}$; indeed, if such a block existed, then the edge $\ell_B c_B^l$ would not belong to G , contradicting Lemma 8.8. It follows that $i < s$, that the edge $w_{i+1}z_{j+1}$ exists, and that the cycle $w_iw_{i+1}z_{j+1}$ delimits an internal face of G . Let v° be the vertex which is different from w_i and such that the cycle $w_{i+1}z_{j+1}v^\circ$ delimits an internal face of G . We have $v^\circ \neq z_{j+2}$, as otherwise z_{j+1} would be transparent, while it is opaque. Also, v° does not belong to a trivial block B that is descendant of B_P , whose parent is w_{i+1} , and such that $\ell_B = z_{j+1}$, as otherwise z_{j+1} would be transparent, while it is opaque. Also, $v^\circ \neq c_P$, since $v^\circ = c_P$ would imply $j+1 = k$,

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

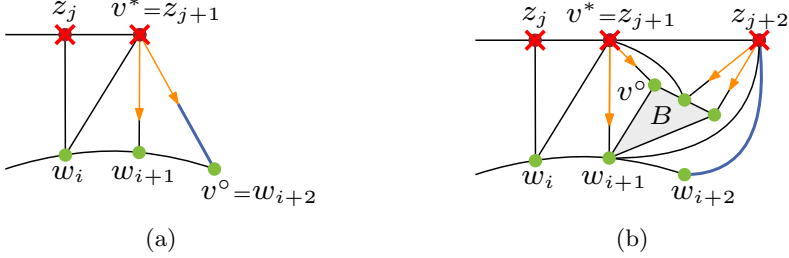


Figure 8.24: Illustrations for Case 13.1 when $v^* = z_{j+1}$, with $j + 1 < k$, and $z_{j+1} \notin \mathcal{I}_P$ (a) if $v^o = w_{i+2}$ and (b) if v^o belongs to a non-trivial pesky block B that is a descendant of B_P and whose parent is w_{i+1} .

given that the edge $c_P z_k$ follows $c_P w_s$ in clockwise order around c_P in G , by construction. Hence, two cases are possible.

- If $v^o = w_{i+2}$ (see Figure 8.24a), then we charge z_{j+1} to w_{i+1} and w_{i+2} and we proceed with $w_{i+2}z_{j+1}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since z_{j+1} has been charged to w_{i+1} and w_{i+2} , which belong to $\mathcal{I}_P^{i+2, j+1}$ and do not belong to $\mathcal{I}_P^{i, j}$. Properties (X2) and (X3) are satisfied since they are satisfied by $w_i z_j$ and since $\mathcal{C}^{i+2, j+1}$ contains the same blocks inside or on its boundary as $\mathcal{C}^{i, j}$. Finally, Property (X4) is satisfied since z_{j+1} is not in $\mathcal{I}_P^{i+2, j+1}$.
- Otherwise, v^o belongs to a non-trivial pesky block B that is a descendant of B_P , whose parent is w_{i+1} , and such that $\ell_B = z_{j+1}$ (see Figure 8.24b). Let $\ell_B = z_{j+1}, z_{j+2}, \dots, z_{j+h} = r_B$ be the cage path of B . By Observation 8.5, we can charge $z_{j+1}, z_{j+2}, \dots, z_{j+h}$ to the $2h$ vertices of B (including w_{i+1}). By Lemma 8.8, the edge $w_{i+2}z_{j+h}$ exists and the cycle $w_{i+1}w_{i+2}z_{j+h}$ delimits an internal face of G . If $j+h = k$, we are done. Otherwise, we proceed with $w_{i+2}z_{j+h}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since each of $z_{j+1}, z_{j+2}, \dots, z_{j+h}$ has been charged to two distinct vertices in B , which belong to $\mathcal{I}_P^{i+2, j+h}$ and do not belong to $\mathcal{I}_P^{i, j}$. Property (X2) is satisfied since it is satisfied by $w_i z_j$. Property (X3) is satisfied by setting $\phi_2 = w_{i+2}$. Finally, Property (X4) is satisfied since z_{j+h} is not in $\mathcal{I}_P^{i+2, j+h}$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

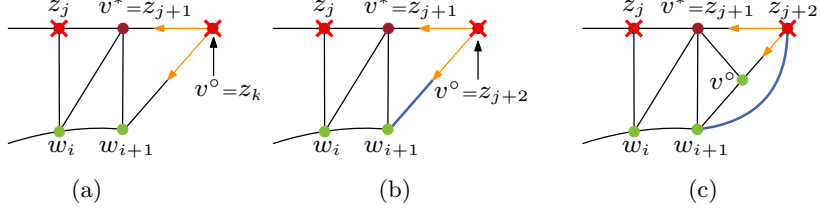


Figure 8.25: Illustrations for Case 13.1 when $v^* = z_{j+1}$, with $j+1 < k$, and $z_{j+1} \in \mathcal{I}_P$ (a) if $v^\circ = z_{j+2}$ and $j+2 = k$, (b) if $v^\circ = z_{j+2}$ and $j+2 < k$, and (c) if v° belongs to a trivial block B that is a descendant of B_P and whose parent is w_{i+1} .

(IV) Fourth, suppose that $v^* = z_{j+1}$, with $j+1 < k$, and that $z_{j+1} \in \mathcal{I}_P$ (see Figure 8.25). Then z_{j+1} is transparent and the cycle $w_i w_{i+1} z_{j+1}$ delimits an internal face of G . Let v° be the vertex which is different from w_i and such that the cycle $w_{i+1} z_{j+1} v^\circ$ delimits an internal face of G . Since z_{j+1} is transparent, we have that $z_{j+2} \notin \mathcal{I}_P$ and three cases are possible.

- If $v^\circ = z_{j+2}$ and $j+2 = k$ (see Figure 8.25a), we conclude the algorithm by charging z_k to w_{i+1} and z_{j+1} (note that ϕ_1 and ϕ_2 remain uncharged).
- If $v^\circ = z_{j+2}$ and $j+2 < k$ (see Figure 8.25b), then we charge z_{j+2} to w_{i+1} and z_{j+1} , and we proceed with $w_{i+1} z_{j+2}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since z_{j+2} has been charged to w_{i+1} and z_{j+1} , which belong to $\mathcal{I}_P^{i+1, j+2}$ and not $\mathcal{I}_P^{i, j}$. Properties (X2) and (X3) are satisfied since they are satisfied by $w_i z_j$ and since $\mathcal{C}^{i+1, j+2}$ contains the same blocks inside or on its boundary as $\mathcal{C}^{i, j}$. Finally, Property (X4) is satisfied since z_{j+2} is not in $\mathcal{I}_P^{i+1, j+2}$.
- Otherwise, v° belongs to a trivial block B that is a descendant of B_P , whose parent is w_{i+1} , and such that $\ell_B = z_{j+1}$ and $r_B = z_{j+2}$ (see Figure 8.25c). We charge z_{j+2} to v° and z_{j+1} and we proceed with $w_{i+1} z_{j+2}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since z_{j+2} has been charged to v° and z_{j+1} , which belong to $\mathcal{I}_P^{i+1, j+2}$ and not to $\mathcal{I}_P^{i, j}$. Property (X2) is satisfied since it is satisfied by $w_i z_j$. Property (X3) is satisfied by setting $\phi_2 = w_{i+1}$. Finally, Property (X4) is satisfied since z_{j+2} is not in $\mathcal{I}_P^{i+1, j+2}$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

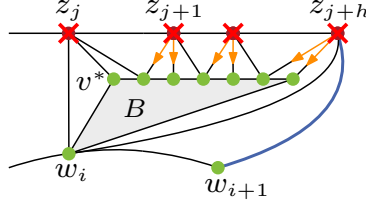


Figure 8.26: Illustration for Case 13.1 when v^* belongs to a non-trivial pesky block B that is a descendant of B_P and whose parent is w_i .

- (V) Fifth, suppose that v^* belongs to a non-trivial pesky block B that is a descendant of B_P , whose parent is w_i , and such that $\ell_B = z_j$ (see Figure 8.26). Let $\ell_B = z_j, z_{j+1}, \dots, z_{j+h} = r_B$ be the cage path of B . By Observation 8.5, we can charge $z_{j+1}, z_{j+2}, \dots, z_{j+h}$ to $2h$ vertices among the $2h + 1$ vertices of B different from w_i . By Lemma 8.8, the edge $w_{i+1}z_{j+h}$ exists and the cycle $w_i w_{i+1} z_{j+h}$ delimits an internal face of G . If $j + h = k$, we are done; note that this is always the case if w_{i+1} is actually c_P , given that the edge $c_P z_k$ follows $c_P w_s$ in clockwise order around c_P in G , by construction. Otherwise, we proceed with $w_{i+1}z_{j+h}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since each of $z_{j+1}, z_{j+2}, \dots, z_{j+h}$ has been charged to two distinct vertices in B different from w_i ; these vertices belong to $\mathcal{I}_P^{i+1, j+h}$ and do not belong to $\mathcal{I}_P^{i, j}$. Property (X2) is satisfied since it is satisfied by $w_i z_j$. Also, Property (X3) is satisfied by setting $\phi_2 = w_{i+1}$; note that there is even one more vertex of B which has not been charged with any vertex. Finally, Property (X4) is satisfied since z_{j+h} is not in $\mathcal{I}_P^{i+1, j+h}$.
- (VI) Sixth, suppose that v^* belongs to a trivial block B that is a descendant of B_P , whose parent is w_i , and such that $\ell_B = z_j$. Let h be the maximal index such that, for $l = 0, 1, \dots, h$, there exists a trivial block B_{i+l} that is a descendant of B_P , whose parent is w_{i+l} , and such that $\ell_{B_{i+l}} = z_{j+l}$. Note that $h \geq 0$, since $B_i = B$ satisfies the required properties. Also, for $l = 0, 1, \dots, h - 1$, we have $r_{B_{i+l}} = z_{j+l+1} = \ell_{B_{i+l+1}}$ and such a vertex is opaque. Moreover, we have that $r_{B_{i+h}} = z_{j+h+1}$; such a vertex might be opaque or transparent (see Figure 8.27 for an example with $h = 2$). For $l = 0, \dots, h$, let d_{i+l} be the vertex of B_{i+l} different from w_{i+l} . For $l = 1, \dots, h$, we charge z_{j+l} to w_{i+l} and to d_{i+l-1} . If $j + h + 1 = k$ (see Figure 8.28a), then we conclude the algorithm by charging z_k to ϕ_1

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

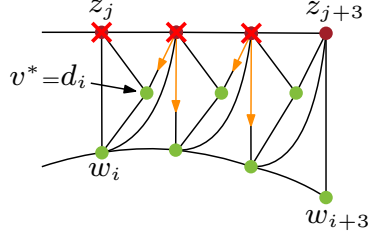


Figure 8.27: Illustration for Case 13.1 when v^* belongs to a trivial block B that is a descendant of B_P , whose parent is w_i , and assuming $h = 2$. The figure only shows the charging of z_{j+1} and z_{j+2} .

and to d_{i+h} . Thus we can assume that $j + h + 1 < k$. This implies that $i + h < s$, given that the edge $c_P z_k$ follows $c_P w_s$ in clockwise order around c_P in G , by construction. By Lemma 8.8, the edge $w_{i+h+1} z_{j+h+1}$ exists and the cycle $w_{i+h} w_{i+h+1} z_{j+h+1}$ delimits an internal face of G . Note that z_{j+h+1} still needs to be charged, if it is opaque, and that d_{i+h} and w_{i+h+1} have not yet been charged with any vertex. We cannot simply proceed with $w_{i+h+1} z_{j+h+1}$ as the active edge, even if z_{j+h+1} is opaque, as Property (X3) would not be satisfied. Indeed, we encountered a block B descendant of B_P but we could not set ϕ_2 to any uncharged vertex. For this reason, we need to perform a case distinction very similar to the one we just presented, but with one more guarantee: Let $w_{i+h+1} z_{j+h+1} u^*$ be the cycle delimiting an internal face of G with $u^* \neq w_{i+h}$; then, by the maximality of h , we have that u^* does not belong to a trivial block that is a descendant of B_P , whose parent is w_{i+h+1} , and such that $\ell_B = z_{j+h+1}$. Note that $u^* \neq c_P$, as $u^* = c_P$ would imply that $j + h + 1 = k$, given that the edge $c_P z_k$ follows $c_P w_s$ in clockwise order around c_P in G , by construction. Thus, the following cases arise.

1. First, if $u^* = w_{i+h+2}$ (see Figure 8.28b), then z_{j+h+1} is opaque. We charge z_{j+h+1} to w_{i+h+1} and to d_{i+h} , and we proceed with $w_{i+h+2} z_{j+h+1}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since each of $z_{j+1}, z_{j+2}, \dots, z_{j+h+1}$ has been charged to two distinct vertices that belong to $\mathcal{I}_P^{i+h+2, j+h+1}$ and do not belong to $\mathcal{I}_P^{i, j}$. Property (X2) is satisfied since it is satisfied by $w_i z_j$. Also, Property (X3) is satisfied by setting $\phi_2 = w_{i+h+2}$. Finally, Property (X4) is satisfied since z_{j+h+1} is opaque and hence

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

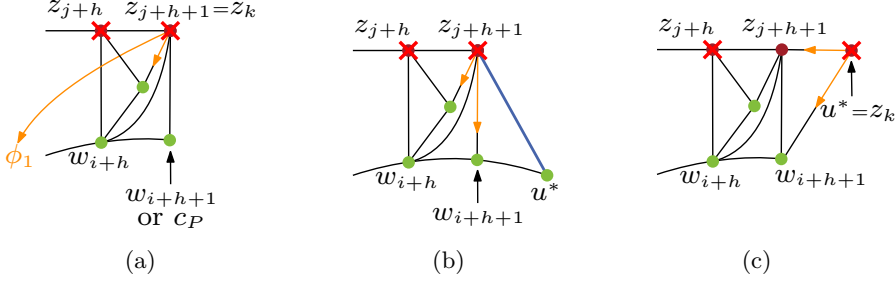


Figure 8.28: Illustrations for Case 13.1 when v^* belongs to a trivial block B that is a descendant of B_P and whose parent is w_i (a) if $j + h + 1 = k$, (b) if $j + h + 1 < k$ and $u^* = w_{i+h+2}$, and (c) if $u^* = z_{j+h+2} = z_k$.

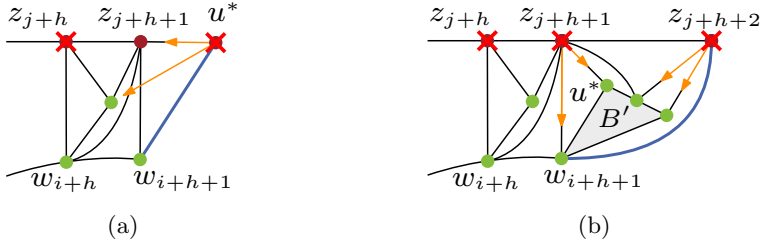


Figure 8.29: Illustrations for Case 13.1 when v^* belongs to a trivial block B that is a descendant of B_P and whose parent is w_i (a) if $u^* = z_{j+h+2}$, with $j+h+2 < k$ and (b) if u^* belongs to a non-trivial pesky block B' that is a descendant of B_P and whose parent is w_{i+h+1} .

it is not in $\mathcal{I}_P^{i+h+2, j+h+1}$.

2. Second, if $u^* = z_{j+h+2} = z_k$ (see Figure 8.28c), then z_{j+h+1} is transparent and belongs to $\mathcal{I}_P$. We conclude the algorithm by charging z_k to w_{i+h+1} and z_{j+h+1} (note that ϕ_1 and d_{i+h} remain unchanged).
3. Third, suppose that $u^* = z_{j+h+2}$, with $j+h+2 < k$ (see Figure 8.29a). Then z_{j+h+1} is transparent and belongs to $\mathcal{I}_P$, while z_{j+h+2} does not belong to $\mathcal{I}_P$. We charge z_{j+h+2} to d_{i+h} and to z_{j+h+1} , and we proceed with $w_{i+h+1}z_{j+h+2}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since each of $z_{j+1}, z_{j+2}, \dots, z_{j+h}, z_{j+h+2}$ has been charged to two distinct vertices that belong to

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

$\mathcal{I}_P^{i+h+1, j+h+2}$ and do not belong to $\mathcal{I}_P^{i, j}$. Property (X2) is satisfied since it is satisfied by $w_i z_j$. Also, Property (X3) is satisfied by setting $\phi_2 = w_{i+h+1}$. Finally, Property (X4) is satisfied since z_{j+h+2} is not in $\mathcal{I}_P^{i+h+1, j+h+2}$.

4. Finally, suppose that u^* belongs to a non-trivial pesky block B' that is a descendant of B_P , whose parent is w_{i+h+1} , and such that $\ell_{B'} = z_{j+h+1}$ (see [Figure 8.29b](#)). Let $\ell_{B'} = z_{j+h+1}, z_{j+h+2}, \dots, z_{j+h+h'} = r_{B'}$ be the cage path of B' . By [Observation 8.5](#), we can charge $z_{j+h+1}, z_{j+h+2}, \dots, z_{j+h+h'}$ to the $2h'$ vertices of B' , including w_{i+h+1} . If $j+h+h' = k$, we are done. Otherwise, we proceed with $w_{i+h+1} z_{j+h+h'}$ as an active edge. Property (X1) is satisfied since it is satisfied by $w_i z_j$ and since each of $z_{j+1}, z_{j+2}, \dots, z_{j+h+h'}$ has been charged to two distinct vertices that belong to $\mathcal{I}_P^{i+h+1, j+h+h'}$ and do not belong to $\mathcal{I}_P^{i, j}$. Property (X2) is satisfied since it is satisfied by $w_i z_j$. Also, Property (X3) is satisfied by setting $\phi_2 = d_{i+h}$. Finally, Property (X4) is satisfied since $z_{j+h+h'}$ is not in $\mathcal{I}_P^{i+h+1, j+h+h'}$.

This concludes the discussion of Case 13.1.

In **Case 13.2**, z_k is transparent and z_{k-1} is opaque. Then, for $j = k-2, k-1, \dots, 2$, if z_j is a transparent vertex, it is in $\mathcal{I}_P$ if $z_{j+1} \notin \mathcal{I}_P$, and it is not in $\mathcal{I}_P$ if $z_{j+1} \in \mathcal{I}_P$. We have that $\mathcal{I}_P$ satisfies Property C. The proof is symmetric to the one of Case 13.1.

In **Case 13.3**, we have that Cases 13.1 and 13.2 do not apply, and that z_k is opaque. Then, for $j = 2, \dots, k-1$, if z_j is a transparent vertex, it is in $\mathcal{I}_P$ if $z_{j-1} \notin \mathcal{I}_P$, and it is not in $\mathcal{I}_P$ if $z_{j-1} \in \mathcal{I}_P$.

We prove that $\mathcal{I}_P$ satisfies Property A. The proof that $G[\mathcal{I}_P \cup \{c_P\}]$ is an outerplane graph is very similar to the one of Case 13.1, with the only difference in the argument which proves that, for any two transparent vertices z_j and z_{j+1} , at most one of them belongs to $\mathcal{I}_P \cup \{c_P\}$. This comes from the algorithm's construction if $2 \leq j \leq k-2$ and from the fact that ℓ_P and r_P are not in $\mathcal{I}_P \cup \{c_P\}$ if $j = 1$ or $j = k-1$, respectively.

We prove that $|\mathcal{I}_P| \geq 2(n_P - 1)/3$. As in Case 13.1, we iteratively charge the vertices $z_1, \dots, z_k$ that do not belong to $\mathcal{I}_P$ to two vertices in $\mathcal{I}_P$, so that each vertex in $\mathcal{I}_P$ is charged with at most one vertex. We again maintain an active edge $w_i z_j$, for which $\mathcal{C}^{i, j}$ and $\mathcal{I}_P^{i, j}$ are defined as in Case 13.1, so that the following properties are satisfied:

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

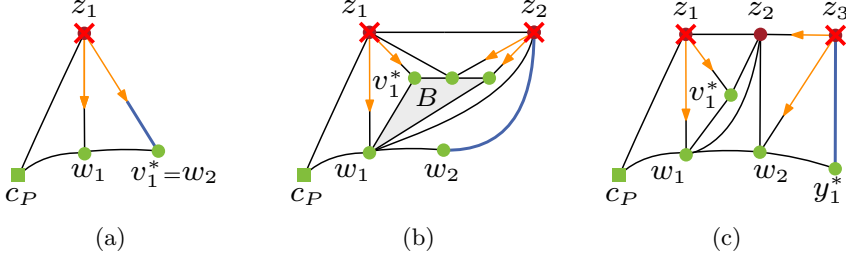


Figure 8.30: Illustrations for Case 13.3 (a) if $v_1^* = w_2$, (b) if v_1^* belongs to a non-trivial pesky block B that is a descendant of B_P and whose parent is w_1 , and (c) if v_1^* belongs to a trivial block B that is a descendant of B_P and whose parent is w_1 .

- (Y1) All the vertices among $z_1, \dots, z_j$ that are not in $\mathcal{I}_P^{i,j}$ have been charged to two vertices in $\mathcal{I}_P^{i,j}$, so that each vertex in $\mathcal{I}_P^{i,j}$ has been charged with at most one vertex.
- (Y2) If $j < k$ and if all the vertices of a block B that is a descendant of B_P are inside or on the boundary of $\mathcal{C}^{i,j}$, then $\mathcal{I}_P^{i,j}$ contains a *first free vertex* ϕ_1 , that is, a vertex that has not been charged with any vertex among $z_1, \dots, z_j$.
- (Y3) We have that z_j is not in $\mathcal{I}_P^{i,j}$.

Observe that, once $j = k$, Property (Y1) implies that $|\mathcal{I}_P| \geq 2(n_P - 1)/3$.

While the iteration in the charging scheme is extremely similar to the one in Case 13.1, the initialization is much more complicated and distinguishes several cases. Let $w_1 z_1 v_1^*$ be the cycle delimiting an internal face of G with $v^* \neq c_P$.

- First, suppose that $v_1^* = w_2$ (see Figure 8.30a). Then z_1 is opaque. We charge z_1 to w_1 and w_2 and proceed with $w_2 z_1$ as an active edge. Property (Y1) is satisfied since z_1 has been charged to two vertices in $\mathcal{I}_P^{2,1}$ and each of them has been charged with z_1 only. Also, Property (Y2) is vacuously satisfied since $\mathcal{C}^{2,1}$ does not contain inside or on its boundary any block B that is a descendant of B_P . Finally, Property (Y3) is satisfied since z_1 is not in $\mathcal{I}_P^{2,1}$, by construction.
- Second, suppose that v_1^* belongs to a non-trivial pesky block B that is a descendant of B_P , whose parent is w_1 , and such that $\ell_B = z_1$ (see

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

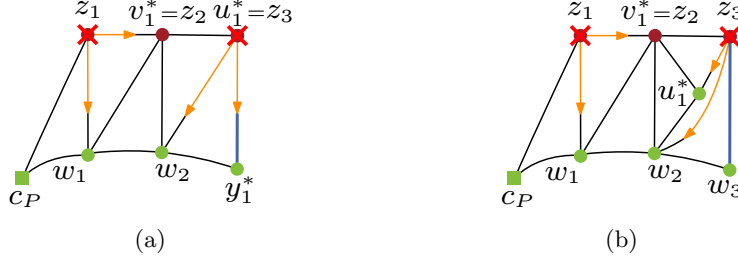


Figure 8.31: Illustrations for Case 13.3 when $v_1^* = z_2$ (a) if $u_1^* = z_3$, and (b) if u_1^* belongs to a trivial block B that is a descendant of B_P and whose parent is w_2 .

Figure 8.30b). Let $\ell_B = z_1, z_2, \dots, z_h = r_B$ be the cage path of B . Note that all of $z_1, z_2, \dots, z_h$ are opaque. By **Observation 8.5**, we can charge $z_1, z_2, \dots, z_h$ to the $2h$ vertices of B , including w_1 . By **Lemma 8.8**, the edge $w_2 z_h$ exists and the cycle $w_1 w_2 z_h$ delimits an internal face of G . If $h = k$, we are done. Otherwise, we proceed with $w_2 z_h$ as an active edge. Property (Y1) is satisfied since each of $z_1, z_2, \dots, z_h$ has been charged to two distinct vertices which belong to $\mathcal{I}_P^{2,h}$. Property (Y2) is satisfied by setting $\phi_1 = w_2$. Also, Property (Y3) is satisfied since z_h is not in $\mathcal{I}_P^{2,h}$.

- Third, suppose that $v_1^* = z_2$ (see **Figure 8.31**). It follows that z_1 is transparent. Since we are not in Case 13.1, we have that z_2 is transparent, as well. This implies that the edge $w_2 z_2$ exists and that the cycle $w_1 w_2 z_2$ delimits an internal face of G . Note that $z_2 \in \mathcal{I}_P$, by construction, and hence $z_3 \notin \mathcal{I}_P$. Let $w_2 z_2 u_1^*$ be the cycle delimiting an internal face of G with $u_1^* \neq w_1$. Since z_2 is transparent, we have that either $u_1^* = z_3$ or that u_1^* belongs to a trivial block B that is a descendant of B_P , whose parent is w_2 , and such that $\ell_B = z_2$. We discuss these two cases.

Suppose first that $u_1^* = z_3$ and let $w_2 z_3 y_1^*$ be the cycle delimiting an internal face of G with $y_1^* \neq z_2$ (see **Figure 8.31a**). By **Lemma 8.8**, we have that y_1^* does not belong to a block B' that is a descendant of B_P , whose parent is w_2 , and such that $\ell_{B'} = z_3$. Also, $y_1^* \neq z_4$, as otherwise z_3 would have degree 3 in G . Hence, we have that $y_1^* = w_3$, then we charge z_1 to w_1 and z_2 , we charge z_3 to w_2 and w_3 , and we proceed with $w_3 z_3$ as an active edge. Note that $k > 3$, since $\mathcal{C}^{3,3}$ does not contain inside or on its boundary all the vertices of any block that is a descendant of B_P . For

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

the same reason, (Y2) is vacuously satisfied. Property (Y1) is satisfied since z_1 and z_3 have each been charged to two vertices in $\mathcal{I}_P^{3,3}$ and each vertex in $\mathcal{I}_P^{3,3}$ has been charged with one vertex only. Also, Property (Y3) is satisfied since z_3 is not in $\mathcal{I}_P^{3,3}$, by construction.

Suppose next that u_1^* belongs to a trivial block B that is a descendant of B_P , whose parent is w_2 , and such that $\ell_B = z_2$ (see Figure 8.31b). Then $r_B = z_3$ and, by Lemma 8.8, the edge w_3z_3 exists and the cycle $w_2w_3z_3$ delimits an internal face of G . We charge z_1 to w_1 and z_2 and we charge z_3 to w_2 and to u_1^* . If $k = 3$, we are done. Otherwise, we proceed with w_3z_3 as an active edge. Property (Y1) is satisfied since z_1 and z_3 have each been charged to two vertices in $\mathcal{I}_P^{3,3}$ and each vertex in $\mathcal{I}_P^{3,3}$ has been charged with one vertex only. Property (Y2) is satisfied by setting $\phi_1 = w_3$. Also, Property (Y3) is satisfied since z_3 is not in $\mathcal{I}_P^{3,3}$.

- Finally, suppose that v_1^* belongs to a trivial block B that is a descendant of B_P , whose parent is w_1 , and such that $\ell_B = z_1$ (see Figure 8.30c). Note that $r_B = z_2$ and that z_1 is transparent. Since we are not in Case 13.1, z_2 is transparent, as well. By Lemma 8.8, the edge w_2z_2 exists and the cycle $w_1w_2z_2$ delimits an internal face of G . Since z_2 is transparent, the edge w_2z_3 exists and the cycle $w_2w_3z_3$ delimits an internal face of G . Let $w_2z_3y_1^*$ be the cycle delimiting an internal face of G with $y_1^* \neq z_2$. By Lemma 8.8, we have that y_1^* does not belong to a block B that is a descendant of B_P , whose parent is w_2 , and such that $\ell_B = z_3$. Also, $y_1^* \neq z_4$, as otherwise z_3 would have degree 3 in G . Hence, $y_1^* = w_3$. We charge z_1 to w_1 and v_1^* , and we charge z_3 to z_2 and w_2 . If $k = 3$, we are done. Otherwise, we proceed with w_3z_3 as an active edge. Property (Y1) is satisfied since z_1 and z_3 have each been charged to two vertices in $\mathcal{I}_P^{3,3}$ and each vertex in $\mathcal{I}_P^{3,3}$ has been charged with one vertex only. Property (Y2) is satisfied by setting $\phi_1 = w_3$. Also, Property (Y3) is satisfied since z_3 is not in $\mathcal{I}_P^{3,3}$.

This concludes the initialization. The algorithm now proceeds iteratively, by means of a case distinction which is the same as in Case 13.1, except for three differences. First, whenever in Case 13.1 we set the second free vertex ϕ_2 after encountering a block B descendant of B_P , here we set the first free vertex ϕ_1 (which in Case 13.1 was already set in the initialization). This guarantees the satisfaction of Property (Y2) throughout the algorithm. Second, in Case 13.1(II), in which $v^* = z_k$, we conclude the algorithm by charging z_k to ϕ_1 and ϕ_2 . Here, we do not have the second free vertex ϕ_2 , however we can exploit the fact

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

that z_k is opaque. Indeed, this implies that $i < s$ and thus we can conclude the algorithm by charging z_k to ϕ_1 and w_s . Third, in Case 13.1(VI), if $j + h + 1 = k$ we conclude the algorithm by charging z_k to ϕ_1 and to d_{i+h} . Here, we have not necessarily set the first free vertex ϕ_1 , however we can exploit the fact that z_k is opaque. Indeed, this implies that $i + h < s$ and thus we can conclude the algorithm by charging z_k to w_s and to d_{i+h} .

In **Case 13.4**, we have that Cases 13.1 and 13.2 do not apply, and that z_1 is opaque. Then, for $j = k - 1, \dots, 2$, if z_j is a transparent vertex, it is in $\mathcal{I}_P$ if $z_{j+1} \notin \mathcal{I}_P$, and it is not in $\mathcal{I}_P$ if $z_{j+1} \in \mathcal{I}_P$. We have that $\mathcal{I}_P$ satisfies Property A. The proof is symmetric to the one of Case 13.3.

If we are in neither of Cases 13.1, 13.2, 13.3, and 13.4, then we are in **Case 13.5**, and z_1, z_2, z_{k-1} , and z_k are all transparent. Let a and b be the maximum and minimum indices, respectively, such that $z_1, \dots, z_a$ and $z_b, \dots, z_k$ are all transparent. Note that $a \geq 2$ and $b \leq k - 1$; also, $a = k$ (and $b = 1$) is possible.

We first get rid of the special case $a = k$ (and $b = 1$). If k is odd, we insert in $\mathcal{I}_P$ all the vertices z_j with j even. Then $\mathcal{I}_P$ satisfies Property A. The proof that $G[\mathcal{I}_P \cup \{c_P\}]$ is an outerplane graph is very similar to the one of Case 13.1. In particular, that at most one of any two consecutive transparent vertices z_j and z_{j+1} belongs to $\mathcal{I}_P \cup \{c_P\}$ is true by construction. Also, the number of vertices in $\mathcal{I}_P$ is equal to $\frac{k-1}{2}$ (the number of vertices $z_2, z_4, \dots, z_{k-1}$), plus $k - 1$ (the number of vertices in B_P different from c_P), plus $n_P - 2k$ (the number of vertices not in B_P in blocks descendant of B_P); note that $n_P - 2k \geq 1$, since a (trivial) block that is descendant of B_P exists. Hence, we have $|\mathcal{I}_P| = \frac{k-1}{2} + (k - 1) + (n_P - 2k) \geq \frac{2}{3}(n_P - 1)$, where the inequality is satisfied because $n_P \geq 2k + 1$ and $k \geq 3$. If k is even, we insert in $\mathcal{I}_P$ all the vertices z_j with j even, except for z_k . Then $\mathcal{I}_P$ satisfies Property B. The proof that $G[\mathcal{I}_P \cup \{c_P, \ell_P\}]$ is an outerplane graph is again very similar to the one of Case 13.1. Also, $|\mathcal{I}_P| = (\frac{k}{2} - 1) + (k - 1) + n_P - 2k$, where the first term accounts for the number of vertices $z_2, z_4, \dots, z_{k-2}$. Hence, we have $|\mathcal{I}_P| \geq \frac{2}{3}(n_P - 2)$, given that $n_P \geq 2k + 1$ and $k \geq 2$.

We can now assume $a \neq k$, which implies that $a < b - 1$. We distinguish three cases.

In *Case 13.5.1*, the length of the sequence $z_b, \dots, z_k$ is even. Then we decide which transparent vertices are in $\mathcal{I}_P$ as follows: z_2 is not in $\mathcal{I}_P$ and, for $j = 3, \dots, k - 1$, if z_j is a transparent vertex, it is in $\mathcal{I}_P$ if $z_{j-1} \notin \mathcal{I}_P$, and it is not in $\mathcal{I}_P$ if $z_{j-1} \in \mathcal{I}_P$. We have that $\mathcal{I}_P$ satisfies Property B. The proof that $G[\mathcal{I}_P \cup \{c_P, \ell_P\}]$ is an outerplane graph is very similar to the one of Case 13.1; while there we have $z_2 \notin \mathcal{I}_P$ because z_2 is opaque, here we

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

have $z_2 \notin \mathcal{I}_P$ by construction. In order to prove that $|\mathcal{I}_P| \geq 2(n_P - 2)/3$ we again charge each vertex of G_P not in $\mathcal{I}_P$ and different from c_P and ℓ_P to two vertices in $\mathcal{I}_P$, so that each vertex in $\mathcal{I}_P$ is charged with at most one vertex, and we again maintain an active edge $w_i z_j$, for which $\mathcal{C}^{i,j}$ and $\mathcal{I}_P^{i,j}$ are defined as in Case 13.1, which satisfies the following properties:

- (Z1) All the vertices among $z_2, \dots, z_j$ that are not in $\mathcal{I}_P^{i,j}$ have been charged to two vertices in $\mathcal{I}_P^{i,j}$, so that each vertex in $\mathcal{I}_P^{i,j}$ has been charged with at most one vertex.
- (Z2) We have that z_j is not in $\mathcal{I}_P^{i,j}$.

Note that here we do not even need to maintain a first free vertex ϕ_1 .

Since z_1 is transparent, the edge $w_1 z_2$ exists and the cycle $w_1 z_1 z_2$ either delimits an internal face of G , or contains inside a trivial block B that is a descendant of B_P , whose parent is w_1 , and such that $\ell_B = z_1$ and $r_B = z_2$. Also, the edge $w_2 z_2$ exists and the cycle $w_1 w_2 z_2$ delimits an internal face of G ; this comes from Lemma 8.8 if $w_1 z_1 z_2$ contains inside a trivial block B as above, and it comes from Lemma 8.8 and from the fact that we are not in Case 2 if $w_1 z_1 z_2$ delimits an internal face of G . Indeed, if $w_1 z_1 z_2$ delimits an internal face of G , let $w_1 z_2 v_1^*$ denote the cycle delimiting an internal face of G with $v_1^* \neq z_1$. Then Lemma 8.8 excludes that v_1^* belongs to a block descendant of B_P , and the fact that we are not in Case 2 excludes that $v_1^* = z_3$, as z_2 would have degree 3 in G . Thus, we can charge z_2 to w_1 and w_2 and initialize the algorithm for the charging scheme with $w_2 z_2$ as an active edge. Properties (Z1) and (Z2) are obviously satisfied. Note that, if a trivial block B inside the cycle $w_1 z_1 z_2$ exists, then its vertex different from w_1 remains uncharged.

The main iteration of the charging algorithm then proceeds as in Case 13.1, with cases that we denote as 13.5(I)–13.5(VI) and that correspond to Cases 13.1(I)–13.1(VI), respectively. However, when the algorithm encounters z_{b-1} , it stops and concludes the charging as described in the following. Let w_m and w_{m+1} be the two neighbors in B_P of z_b . Since b is the smallest index such that $z_b, \dots, z_k$ are all transparent and since $a < b - 1$, we have that z_{b-1} exists and is opaque. Consider the step of the algorithm's execution in which z_{b-1} is first encountered. Let $w_i z_j$ be the active edge before that step. Hence, z_{b-1} is not inside or on the boundary of $\mathcal{C}^{i,j}$, but it is inside or on the boundary of $\mathcal{C}^{i',j'}$ if $w_{i'} z_{j'}$ denotes the active edge after $w_i z_j$. Let $w_i z_j v^*$ be the cycle delimiting an internal face of G that does not lie inside $\mathcal{C}^{i,j}$.

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

- The algorithm cannot be in Case 13.5(I), as in this case a vertex $v^* = w_{i+1}$ is encountered.
- The algorithm cannot be in Case 13.5(II), as in this case the vertex $v^* = z_k$ is encountered, and we have $b - 1 < k$.
- The algorithm might be in Case 13.5(III). Indeed, we might have $v^* = z_{j+1} = z_{b-1}$, where $z_{j+1}w_iw_{i+1}$ and $z_{j+1}w_{i+1}w_{i+2}$ are cycles delimiting internal faces of G , as in [Figure 8.24a](#), and $i + 2 \leq m$. Then we conclude the algorithm by charging z_{b-1} to w_{i+1} and w_{i+2} , and by charging z_{b+j} to z_{b+j-1} and w_{m+j} , for $j = 1, 3, 5, \dots, k - b$. It is also possible that the algorithm first encounters z_{b-1} in Case 13.5(III) if $z_{j+1}w_iw_{i+1}$ is a cycle delimiting an internal face of G , and if the vertex $v^\circ \neq w_i$ such that the cycle $w_{i+1}z_{j+1}v^\circ$ delimits an internal face of G belongs to a non-trivial pesky block B , as in [Figure 8.24b](#). In this case, since z_b is transparent and all the vertices of the cage path of B are opaque, z_{b-1} might be the right cage vertex r_B of B . Then the vertices of the cage path of B , including z_{b-1} , can be charged to the vertices of B and we conclude the algorithm by charging z_{b+j} to z_{b+j-1} and w_{m+j} , for $j = 1, 3, 5, \dots, k - b$.
- The algorithm might be in Case 13.5(IV). Indeed, we might have $v^* = z_{j+1}$ and $z_{j+2} = z_{b-1}$, where $z_{j+1} \in \mathcal{I}_P$, where $z_{j+1}w_iw_{i+1}$ and $z_{j+1}z_{j+2}w_{i+1}$ are cycles of G , as in [Figures 8.25b](#) and [8.25c](#), with $i + 1 \leq m$. Then we conclude the algorithm by charging z_{j+2} to w_{i+1} and z_{j+1} , and by charging z_{b+j} to z_{b+j-1} and w_{m+j} , for $j = 1, 3, 5, \dots, k - b$.
- The algorithm might be in Case 13.5(V). Indeed, v^* might belong to a non-trivial pesky block B , as in [Figure 8.26](#), with $z_{b-1} = r_B$. Then the vertices of the cage path of B , including z_{b-1} and excluding ℓ_B , can be charged to the vertices of B different from w_i and we conclude the algorithm by charging z_{b+j} to z_{b+j-1} and w_{m+j} , for $j = 1, 3, 5, \dots, k - b$.
- Finally, the algorithm might be in Case 13.5(VI). Recall that, in this case, v^* belongs to a trivial block B that is a descendant of B_P , whose parent is w_i , and such that $\ell_B = z_j$, as in [Figure 8.27](#). This results in a sequence $B_i, \dots, B_{i+h}$ of trivial blocks descendants of B_P such that, for $l = 0, 1, \dots, h$, the block B_{i+l} has w_{i+l} as parent, has d_{i+l} as vertex different from w_{i+l} , has $\ell_{B_{i+l}} = z_{j+l}$, and has $r_{B_{i+l}} = z_{j+l+1}$. Since the vertices $z_{j+1}, \dots, z_{j+h}$ are opaque, we might have $z_{b-1} = z_{j+h}$ or $z_{b-1} = z_{j+h+1}$. In the former (latter) case, for $l = 1, \dots, h$ (resp. for $l = 1, \dots, h + 1$), we charge z_{j+l} to w_{i+l} and to d_{i+l-1} , and we

8. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN 2-OUTERPLANAR GRAPHS

conclude the algorithm by charging z_{b+j} to z_{b+j-1} and w_{m+j} , for $j = 1, 3, 5, \dots, k - b$. The algorithm might also first encounter z_{b-1} in Case 13.5(VI) if the cycle $w_{i+h}w_{i+h+1}z_{j+h+1}$ delimits an internal face of G and the cycle $w_{i+h+1}z_{j+h+1}u^*$ delimiting an internal face of G with $u^* \neq w_{i+h}$ has u^* in a non-trivial pesky block B' that is a descendant of B_P , whose parent is w_{i+h+1} , such that $\ell_{B'} = z_{j+h+1}$, as in Figure 8.29b, and such that $r_{B'} = z_{b-1}$. Then the vertices of the cage path of B' can be charged to the vertices of B' , and we conclude the algorithm by charging z_{b+j} to z_{b+j-1} and w_{m+j} , for $j = 1, 3, 5, \dots, k - b$.

In Case 13.5.2, the length of the sequence $z_1, \dots, z_a$ is even. This case can be handled symmetrically to Case 13.5.1.

Finally, in Case 13.5.3, we have that a and $k - b + 1$ are both odd. We let $z_2, z_4, \dots, z_{a-1}$ be in $\mathcal{I}_P$ (and $z_3, z_5, \dots, z_a$ not be in $\mathcal{I}_P$). Then, for $j = a + 2, \dots, k - 1$, if z_j is a transparent vertex, it is in $\mathcal{I}_P$ if $z_{j-1} \notin \mathcal{I}_P$, and it is not in $\mathcal{I}_P$ if $z_{j-1} \in \mathcal{I}_P$. We have that $\mathcal{I}_P$ satisfies Property C. The proof that $G[\mathcal{I}_P \cup \{c_P, r_P\}]$ is an outerplane graph is again very similar to the one of the previous cases. Note that $z_{k-1} \notin \mathcal{I}_P$, given that $k - b + 1$ is odd. In order to prove that $|\mathcal{I}_P| \geq 2(n_P - 2)/3$ we again charge each vertex of G_P not in $\mathcal{I}_P$ and different from c_P and r_P to two vertices in $\mathcal{I}_P$, so that each vertex in $\mathcal{I}_P$ is charged with at most one vertex, and we maintain an active edge $w_i z_j$, which satisfies Properties (Z1) and (Z2). Note that, since $a \geq 2$ and a is odd, we have $a \geq 3$. Since z_1, z_2 , and z_3 are transparent, the cycles $w_1 z_1 z_2$, $w_1 w_2 z_2$, $w_2 z_2 z_3$, and $w_2 w_3 z_3$ exist; the second and fourth cycles delimit internal faces of G , while each of the first and third cycles either delimits an internal face of G or contains inside a trivial block descendant of B_P . Regardless, we can charge z_1 to w_1 and z_2 , and z_3 to w_2 and w_3 , and initialize the algorithm for the charging scheme with $w_3 z_3$ as an active edge. Properties (Z1) and (Z2) are obviously satisfied.

The algorithm then proceeds exactly as in Case 13.5.1, except for the fact that, when it encounters and charges z_{b-1} to two vertices in $\mathcal{I}_P$, it terminates by charging z_{b+j} to z_{b+j-1} and w_{i+j} , for $j = 1, 3, 5, \dots, k - b - 1$, rather than for $j = 1, 3, 5, \dots, k - b$; note that $z_k = r_P$ does not need to be charged. $\square$

By employing Lemma 8.11, we can now apply induction in the same way as described after the proof of Lemma 8.10, with $\mathcal{I}_P, c_P, \ell_P, r_P, G_P$ in place of $\mathcal{I}_B, c_B, \ell_B, r_B, G_B$, respectively, so to find a good set for G . This concludes the proof of Theorem 8.1.

Large Induced Outerplanar Subgraphs in Planar 3-Trees

In this chapter¹, we study the maximum size of an induced outerplanar graph that one is guaranteed to find in a planar 3-tree. Planar 3-trees form a fundamental subclass of planar graphs: they are exactly the maximal planar graphs of treewidth 3, and they admit a simple recursive construction that makes them particularly amenable to inductive and structural arguments. Although their bounded treewidth makes polynomial-time algorithms conceivable for many otherwise NP-hard problems [220, 221, 222], relatively little is known about extremal properties of induced outerplanar subgraphs in planar 3-trees.

Chapter contributions. Our contribution is twofold. First, we present a construction showing that there exist n -vertex planar 3-trees in which every induced outerplanar subgraph has size at most $\frac{7}{10}n + \mathcal{O}(1)$. Second, we prove that every n -vertex planar 3-tree contains an induced outerplanar subgraph on at least $\frac{2}{3}n$ vertices.

Related work. Although, to the best of our knowledge, the problem of determining the maximum size of an induced outerplanar subgraph that one is guaranteed to find in a planar 3-tree has not been addressed explicitly before, several results from the literature are related to our work or have implications for our problem.

¹The content of this chapter is part of joint work with Samuel Schneider and Torsten Ueckerdt and was carried out during a visiting period at the Institute of Theoretical Informatics at the Karlsruhe Institute of Technology.

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

Every n -vertex graph with treewidth k contains an induced subgraph with treewidth t on at least $\frac{t+1}{k+1}n$ vertices [215]. For $t = 1$ and $k = 3$, this result yields a lower bound of $\frac{n}{2}$ on the size of an induced forest—which is also outerplanar—that one is guaranteed to find in an n -vertex graph with treewidth 3.

Moreover, the best known lower bound on the size of an induced outerplanar subgraph that one is guaranteed to find in an n -vertex graph with treewidth 2 is $\frac{2}{3}n$, which again follows from the result of [215] by setting $t = 1$ and $k = 2$. We believe that a stronger bound of $\frac{4}{5}n$ is attainable in this case; this bound would be tight, as witnessed by the graph $K_{2,3}$, where at least one vertex must be removed to obtain an outerplanar graph. Even if this bound were correct, it would imply that the size of an induced outerplanar subgraph that one is guaranteed to find in an n -vertex graph with treewidth 3 is $\frac{2}{3} \cdot \frac{4}{5}n = \frac{3}{5}n$, which is still smaller than the lower bound we prove in this chapter.

Chapter organization. The rest of this chapter is organized as follows. In Section 9.1 we provide the preliminaries, and in Section 9.2 we prove both the upper and lower bounds.

9.1 Preliminaries

We refer to Chapter 1 for basic definitions on graphs. Let G be a simple undirected graph. We recall that, for any set $\mathcal{I} \subseteq V(G)$, $G[\mathcal{I}]$ denotes the subgraph of G induced by $\mathcal{I}$. We also recall that a planar drawing is outerplanar if every vertex is incident to the outer face and an outerplanar graph is a graph that admits an outerplanar drawing. A *3-tree* is defined recursively as follows:

- The base graph is a complete graph on 4 vertices.
- Given a 3-tree G , a new vertex may be added by making it adjacent to all vertices of an existing clique of size 3 in G .

Equivalently, an n -vertex graph G is a 3-tree if it can be generated by a sequential addition of vertices and their incident edges in an order $v_1, \dots, v_n$ such that, for each $i > 4$, the vertex v_i has exactly 3 neighbors that precede it in the ordering, and the subgraph of G induced by these neighbors is a clique of size 3. A *planar 3-tree* is a 3-tree that admits a planar embedding. Equivalently, a planar 3-tree is a 3-tree obtained by adding each new vertex to a distinct clique of size 3. A *partial 3-tree* is a subgraph of a 3-tree. Analogously, a *planar partial 3-tree* is a planar graph that is a subgraph of a 3-tree. It can be shown

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

that every planar partial 3-tree with $n \geq 4$ vertices is a subgraph of a planar 3-tree with n vertices. Let G be an n -vertex planar 3-tree. We associate G with a rooted tree T_G , called the *representative tree* of G , which encodes the inductive construction of G . If $n = 3$, then T_G consists of a single node corresponding to the initial triangle. If $n = 4$, then T_G consists of a root representing the initial triangle and a single child representing the unique fourth vertex added to the graph. For $n > 4$, the tree T_G is defined as follows. Every vertex of G , except for the three vertices of the initial triangle, corresponds to a node of T_G , and each such node has at most three children. Suppose a vertex p is added during the construction of G as the common neighbor of three vertices u, v, w , thereby forming a K_4 . Among the vertices u, v, w , let x be the one that was added last in the inductive construction of G . Then, the node corresponding to p is attached as a child of the node corresponding to x in T_G . We denote the initial triangle of G (similarly of T_G) as the *parent triangle* of G (similarly of T_G). For any vertex $u \in T_G$, let $T_G(u)$ denote the subtree of T_G rooted at u . Let $C(u)$ be the set of children of u in T_G , and let x_u, y_u , and z_u be the three neighbors of u in $T_G \setminus T_G(u)$. If u is the common neighbor of the parent triangle in T_G , then x_u, y_u , and z_u coincide with the vertices in the parent triangle.

A *cutvertex* of a connected graph G is a vertex whose removal disconnects G . A connected graph is *biconnected* if it contains no cutvertex. A *biconnected component* (or *block*) of a graph G is a maximal (in terms of vertices and edges) biconnected subgraph of G .

Finally, we denote by K_k a clique of size k , and by $K_{a,b}$ the complete bipartite graph with partite sets of sizes a and b , i.e., the graph whose vertex set can be partitioned into two sets of sizes a and b , such that every vertex in one set is adjacent to all vertices in the other set, and there are no edges within each partite set.

9.2 Lower and Upper Bounds

In the following, we present our results. We begin by constructing a family of planar 3-trees that establishes the upper bound.

Theorem 9.1. *For every $n \geq 1$, there exists an n -vertex planar 3-tree G with the following property. Consider any set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ is outerplanar. Then $|\mathcal{I}| \leq \frac{7}{10}n + \mathcal{O}(1)$.*

Proof. To prove the statement, we extend the construction shown in [23] to prove an upper bound for linear forests in outerplanar graphs. Let $h = \lfloor \frac{n}{10} \rfloor$

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

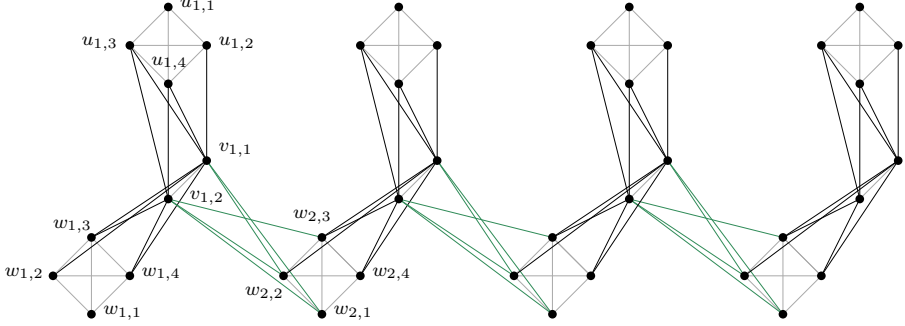


Figure 9.1: Example for [Theorem 9.1](#) when $n = 40$ and $h = 4$. Edges between vertices in the same set are depicted in gray, edges between vertices in different sets of G_i , for $i = 1, \dots, 4$, are depicted in black, and edges between vertices in G_i and G_{i+1} , for $i = 1, \dots, 3$, are depicted in green.

and let G be the n -vertex graph defined as follows (refer to [Figure 9.1](#) for an example when $n = 40$ and $h = 4$). For $i = 1, 2, \dots, h$, G contains two distinct K_4 , denoted by U_i and by W_i , respectively, and one K_2 denoted by V_i . The edges within each U_i , W_i , and V_i are depicted in gray in the example of [Figure 9.1](#). We denote vertices in U_i and W_i by $u_{i,1}, u_{i,2}, u_{i,3}$, and $u_{i,4}$, and by $w_{i,1}, w_{i,2}, w_{i,3}$, and $w_{i,4}$, respectively. We denote vertices in V_i by $v_{i,1}$ and $v_{i,2}$. Then, for $i = 1, \dots, h$, for $j = 1, 2$, and for $l = j + 1, \dots, 4$, we add the edges $(u_{i,l}, v_{i,j})$ and $(w_{i,l}, v_{i,j})$. In the example of [Figure 9.1](#), these edges are depicted in black. Moreover, for $i = 1, \dots, h - 1$, for $j = 1, 2$, and for $l = 1, \dots, j + 1$, we add the edge $(v_{i,j}, w_{i+1,l})$. In the example of [Figure 9.1](#), these edges are depicted in green. Finally, we add $n - 10h$ isolated vertices to G , which do not play any role in the proof, to ensure that it has exactly n vertices. The resulting graph G is a planar 3-tree. Indeed, the above construction can be easily translated into a sequence describing how G is obtained by adding each vertex, i.e., by stacking a different clique, thus producing a planar 3-tree. Note that G is not maximal, but edges between vertices in $V(U_i)$ and in $V(W_i)$ for a fixed i can be added to obtain a maximal planar 3-tree, such that every K_3 in G has at most one common neighbor. Also, for $i = 1, \dots, h$, we define $G_i := G[V(U_i) \cup V(V_i) \cup V(W_i)]$.

In the following, we show that a set $\mathcal{I}' \subseteq V(G)$ with $|\mathcal{I}'| > \frac{7}{10}n$ such that $G[\mathcal{I}']$ is outerplanar cannot exist. First, observe that, for any $i = 1, \dots, h - 1$, $v_{i,1}$ and $v_{i,2}$ form exactly two K_4 with vertices in $V(G_i)$, one with $u_{i,3}$ and $u_{i,4}$,

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

and one with $w_{i,3}$ and $w_{i,4}$. Moreover, if $i < h - 1$, $v_{i,1}$ and $v_{i,2}$ form a K_4 with $w_{i+1,1}$ and $w_{i+1,2}$. Second, for any $i = 1, \dots, h - 1$, it is possible to select a set $\mathcal{I}'_i \subset V(G_i)$, such that $|\mathcal{I}'_i| \leq 8$ and such that $G[\mathcal{I}'_i]$ is outerplanar by excluding at least one vertex in $V(U_i) \setminus \{u_{i,1}\}$ and at least one vertex in $V(W_i) \setminus \{w_{i,1}\}$. This implies that, if $|\mathcal{I}'_i| = 8$, $v_{i,1}$ and $v_{i,2}$ each have exactly two common neighbors in $\mathcal{I}'_i$: one in U_i and one in W_i . Third, for any $i = 1, \dots, h - 1$, it is not possible to select a set $\mathcal{I}'_i \subset V(G_i)$ such that $|\mathcal{I}'_i| > 8$ and $G[\mathcal{I}'_i]$ is outerplanar, since this would require selecting either all vertices in U_i or all vertices in W_i , thus obtaining a K_4 , which is not outerplanar. Fourth, consider any $i = 1, \dots, h - 1$ and a set $\mathcal{I}'_i \subset V(G_i)$ such that $|\mathcal{I}'_i| = 8$ and such that $G[\mathcal{I}'_i]$ is outerplanar. Since $\mathcal{I}'_i$ necessarily contains all vertices in $V(V_i)$, a set $\mathcal{I}'_{i+1} \subset V(G_{i+1})$ such that $G[\mathcal{I}'_i \cup \mathcal{I}'_{i+1}]$ is outerplanar, needs to exclude: (i) at least $w_{i+1,1}$, $w_{i+1,2}$, since otherwise $G[\mathcal{I}'_i \cup \mathcal{I}'_{i+1}]$ would contain $v_{i,1}$ and $v_{i,2}$ with three common neighbors (one in U_i , one in W_i , and one in W_{i+1}), hence a $K_{2,3}$ that is not outerplanar; (ii) one vertex in $V(U_{i+1})$, since otherwise $G[\mathcal{I}'_i \cup \mathcal{I}'_{i+1}]$ would contain a K_4 that is not outerplanar; and (iii) one vertex in $\{v_{i+1,1}, v_{i+1,2}, w_{i+1,3}, w_{i+1,4}\}$, since otherwise $G[\mathcal{I}'_i \cup \mathcal{I}'_{i+1}]$ would contain a K_4 . Thus, $|\mathcal{I}'_{i+1}| \leq 6$. Fifth, consider any $i = 2, \dots, h - 1$ and a set $\mathcal{I}'_i \subset V(G_i)$ such that $|\mathcal{I}'_i| = 8$ and such that $G[\mathcal{I}'_i]$ is outerplanar. Since $|\mathcal{I}'_i| = 8$, as we discussed above, one between $w_{i,3}$ and $w_{i,4}$ must be excluded from $\mathcal{I}'_i$, hence $w_{i,1}$ and $w_{i,2}$ are both in $\mathcal{I}'_i$. This implies that a set $\mathcal{I}'_{i-1} \subset V(G_{i-1})$ such that $G[\mathcal{I}'_i \cup \mathcal{I}'_{i-1}]$ is outerplanar, needs to exclude at least one between $v_{i-1,1}$ and $v_{i-1,2}$, since otherwise $G[\mathcal{I}'_i \cup \mathcal{I}'_{i-1}]$ would contain a K_4 , which is not outerplanar. Hence, $|\mathcal{I}'_{i-1}| \leq 7$. The above arguments show overall that a set $\mathcal{I}' \subseteq V(G)$ with $|\mathcal{I}'| > \frac{7}{10}n$ such that $G[\mathcal{I}']$ is outerplanar cannot exist.

Finally, a set $\mathcal{I} \subseteq V(G)$ with $|\mathcal{I}| \leq \frac{7}{10}n$ such that $G[\mathcal{I}]$ is outerplanar can be obtained by choosing vertices of $V(G)$ that satisfy the following constraints: for each $i = 1, \dots, h$, $|\mathcal{I} \cap V(U_i)| = 3$, $|\mathcal{I} \cap V(W_i)| = 3$, and $|\mathcal{I} \cap V(V_i)| = 1$. This completes the argument. $\square$

We now prove the lower bound, which is formalized in the following theorem.

Theorem 9.2. *For every $n \geq 1$ and for every n -vertex planar 3-tree G , there exists a set $\mathcal{I} \subseteq V(G)$ such that $G[\mathcal{I}]$ is outerplanar and $|\mathcal{I}| \geq \frac{2}{3}n$.*

Proof. Without loss of generality, we can assume that G is a maximal planar 3-tree, as otherwise edges can be added to it so that it becomes maximal, and then a subset $\mathcal{I}$ with the properties required by [Theorem 9.2](#) in the augmented graph satisfies the same properties in the original graph. The proof is by induction on n . In the base case we have $n \leq 5$. It is easy to see that we can

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

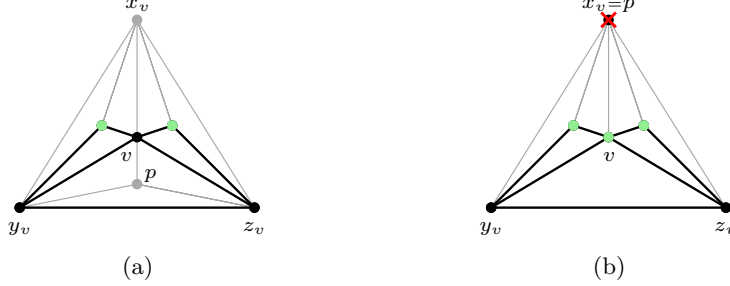


Figure 9.2: Illustrations for (a) Case 1 and (b) Case 2. In this and the following figures, vertices in $\mathcal{I}'$ and in $\mathcal{I}$ are depicted in black, vertices in $\mathcal{I}'$ but not in $\mathcal{I}$ are depicted with a red cross, vertices not in $\mathcal{I}'$ but in $\mathcal{I}$ are depicted in green, and vertices not in $\mathcal{I}'$ and not in $\mathcal{I}$ are depicted in gray. The induced subgraph $G[\mathcal{I}]$ is shown with bold edges.

always find a set $\mathcal{I}$ with n vertices (if $1 \leq n \leq 3$) and with $n - 1$ vertices (if $4 \leq n \leq 5$), such that $G[\mathcal{I}]$ is outerplanar and such that $\frac{|\mathcal{I}|}{n} \geq \frac{2}{3}$.

Consider the representative tree T_G of G . Since we assume $n \geq 6$, there exists at least one vertex u in G such that either: (i) u is the common neighbor of the parent triangle in T_G and has at least two children that are leaves, or (ii) u is not the common neighbor of the parent triangle in T_G and has at least one child that is a leaf. Now consider any vertex v whose children are all leaves. We distinguish three cases according to $|C(v)|$. In the following, we refer to [Figures 9.2 to 9.5](#), which illustrate some of the cases considered in the induction.

Case 1: v has 3 children. We apply the induction on the graph $G' := G[V \setminus C(v)]$. Let $\mathcal{I}'$ be the vertex set obtained by induction such that $G[\mathcal{I}']$ is outerplanar. Without loss of generality, we may assume that $v \in \mathcal{I}'$ and that at least one of x_v, y_v , or z_v is not in $\mathcal{I}'$, say x_v (see [Figure 9.2a](#)). We define $\mathcal{I} := \mathcal{I}' \cup C(v) \setminus \{p\}$, where p is the common neighbor of v, y_v , and z_v . We prove that $G[\mathcal{I}]$ is outerplanar. Indeed, each of the two vertices added from $C(v) \setminus \{p\}$ has degree 2 in $G[\mathcal{I}]$, being adjacent only to v and to one of $\{y_v, z_v\}$. Any path between y_v and z_v that goes through one of these vertices must therefore pass through v , so the number of internally disjoint paths between y_v and z_v does not increase. Hence, $G[\mathcal{I}]$ is outerplanar. Also, since $|V(G)| = |V(G')| + 3$ and $|\mathcal{I}| \geq |\mathcal{I}'| + 2$, it follows that $\frac{|\mathcal{I}|}{|V(G)|} \geq \frac{2}{3}$. Therefore, we can assume that T_G contains no vertex v with exactly three children that are leaves.

Case 2: v has 2 children. We apply the induction on the graph $G' :=$

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

$G[V \setminus (\{v\} \cup C(v))]$. Let $\mathcal{I}'$ be the vertex set obtained by induction such that $G[\mathcal{I}']$ is outerplanar. Without loss of generality, we may assume that x_v, y_v , and z_v all belong to $\mathcal{I}'$ (see [Figure 9.2b](#)). Let p be the vertex among x_v, y_v , and z_v that is adjacent to both vertices in $C(v)$, and define $\mathcal{I} := (\mathcal{I}' \setminus \{p\}) \cup (C(v) \cup \{v\})$. We prove that $G[\mathcal{I}]$ is outerplanar. Indeed, the cycle $y_v x_v z_v$ in $G[\mathcal{I}']$ is replaced by the cycle obtained by removing p and by adding v . Moreover, each of the two vertices added from $C(v)$ has degree 2 in $G[\mathcal{I}]$, and any path between the two vertices in $\{x_v, y_v, z_v\}$ that are not p and that passes through vertices in $C(v)$ must go through v . Overall, this does not increase the number of internally disjoint paths between the two vertices in $\{x_v, y_v, z_v\}$ that are not p . Hence, $G[\mathcal{I}]$ is outerplanar. Also, since $|V(G)| = |V(G')| + 3$ and $|\mathcal{I}| \geq |\mathcal{I}'| + 2$, it follows that $\frac{|\mathcal{I}|}{|V(G)|} \geq \frac{2}{3}$. Thus, we can assume that T_G contains no vertex v with exactly two children that are leaves.

Case 3: v has 1 child. Consider any vertex $w \in T_G$ such that $T_G(w)$ contains at least one vertex v with exactly one child, which is a leaf. Observe that, under the current assumptions, w can have at most two other children, each of which is either a leaf or a vertex with one child that is a leaf. We distinguish five different cases depending on the structure of $T_G(w)$. These correspond to the ways of assigning the values $\{0, 1, 2\}$ into a 3-tuple $\langle a, b, c \rangle$, up to symmetry, such that at least one element of the tuple is equal to 2. In the following, recall that for a vertex w , we denote by x_w, y_w , and z_w the three neighbors of w in $T_G \setminus T_G(w)$. Furthermore, we denote by v_1 the child of w that is adjacent to both x_w and y_w , by v_2 the child of w that is adjacent to both x_w and z_w , and by v_3 the child of w that is adjacent to both y_w and z_w . With a slight abuse of notation, when defining the graph to which we apply induction, we also use $T_G(w)$ to denote the set of vertices in the subtree of T_G rooted at w .

Case 3.1: $\langle 2, 2, 2 \rangle$. In this case, w has three children v_1, v_2 , and v_3 in T_G , each of which has a child u_1, u_2 , and u_3 , respectively. We apply the induction on the graph $G' := G[(V \setminus T_G(w)) \cup \{w\}]$. Let $\mathcal{I}'$ be the vertex set obtained by induction such that $G[\mathcal{I}']$ is outerplanar. Without loss of generality, we may assume that $w \in \mathcal{I}'$ and that one of x_w, y_w , and z_w is not in $\mathcal{I}'$, say x_w (see [Figure 9.3a](#)). When a different vertex in $\{y_w, z_w\}$ is not in $\mathcal{I}'$, we can proceed in a similar way. We define $\mathcal{I} := (\mathcal{I}' \setminus \{w\}) \cup \{v_1, v_2, v_3, u_1, u_2\}$. We prove that $G[\mathcal{I}]$ is outerplanar, assuming $x_w \notin \mathcal{I}'$. The argument is analogous when a different vertex in $\{y_w, z_w\}$ is not in $\mathcal{I}'$. Indeed, the cycle $y_w w z_w$ in $G[\mathcal{I}']$ is replaced by the cycle $y_w v_3 z_w$, which does not increase the number of internally disjoint paths between y_w and z_w . The two sets of vertices $\{v_1, u_1\}$ and $\{v_2, u_2\}$

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

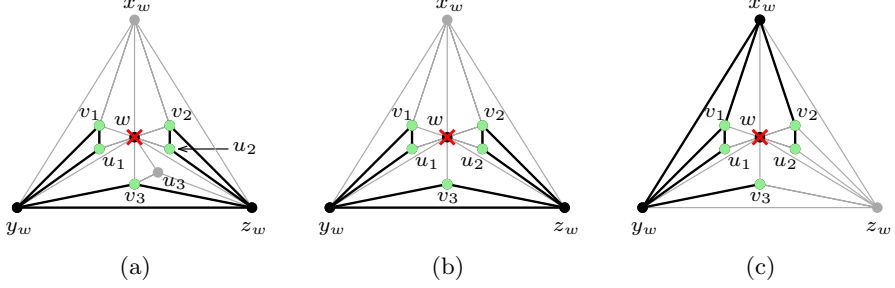


Figure 9.3: Illustrations for (a) Case 3.1 and (b)-(c) Case 3.2.

form two biconnected components in $G[\mathcal{I}]$ with y_w and z_w , respectively (which are either cycles of length 3 or paths of length 2 depending on the neighbors of u_1 and u_2), such that y_w and z_w are cutvertices. Moreover, v_3 has only y_w and z_w as neighbors in $G[\mathcal{I}]$ (observe that v_3 has also u_3 as neighbor in G , but u_3 is not included in both $\mathcal{I}'$ and $\mathcal{I}$). Hence, this does not increase the number of internally disjoint paths between y_w and z_w . Thus, $G[\mathcal{I}]$ is outerplanar. Moreover, since $|V(G)| = |V(G')| + 6$ and $|\mathcal{I}| \geq |\mathcal{I}'| + 4$, it follows that $\frac{|\mathcal{I}|}{|V(G)|} \geq \frac{2}{3}$. Note that, in this case, the exact neighbors of u_1 , u_2 , and u_3 , that is, the vertices forming a K_4 with them, are not relevant for the argument.

Case 3.2: $\langle 2, 2, 1 \rangle$. In this case, w has three children v_1, v_2 , and v_3 in T_G , where v_1 and v_2 have children u_1 and u_2 , respectively. We apply the induction on the graph $G' := G[(V \setminus T_G(w)) \cup \{w\}]$. Let $\mathcal{I}'$ be the vertex set obtained by induction such that $G[\mathcal{I}']$ is outerplanar. Without loss of generality, we may assume that $w \in \mathcal{I}'$ and that one of x_w, y_w , and z_w is not in $\mathcal{I}'$. We distinguish two subcases, depending on which of x_w and z_w is not in $\mathcal{I}'$ (the case where y_w is not in $\mathcal{I}'$ is symmetric to the latter). In the former subcase (see Figure 9.3b), we define $\mathcal{I} := (\mathcal{I}' \setminus \{w\}) \cup \{v_1, v_2, v_3, u_1, u_2\}$. In the latter subcase, we define $\mathcal{I} := (\mathcal{I}' \setminus \{p\}) \cup \{v_1, v_2, v_3, u_1, u_2\}$, where $p = x_w$ if u_1 is a neighbor of x_w , and $p = w$ otherwise (see Figure 9.3c when $p = w$). We prove that $G[\mathcal{I}]$ is outerplanar. Indeed, in the first subcase, the cycle $y_w w z_w$ in $G[\mathcal{I}']$ is replaced by the cycle $y_w v_3 z_w$, which does not increase the number of internally disjoint paths between y_w and z_w . The two sets of vertices $\{v_1, u_1\}$ and $\{v_2, u_2\}$ form two biconnected components in $G[\mathcal{I}]$ with y_w and z_w , respectively (which are either cycles of length 3 or paths of length 2 depending on the neighbors of u_1 and u_2), such that y_w and z_w are cutvertices. Also, v_3 has only y_w and z_w as neighbors in $G[\mathcal{I}]$. Hence, this does not increase the number of internally disjoint

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

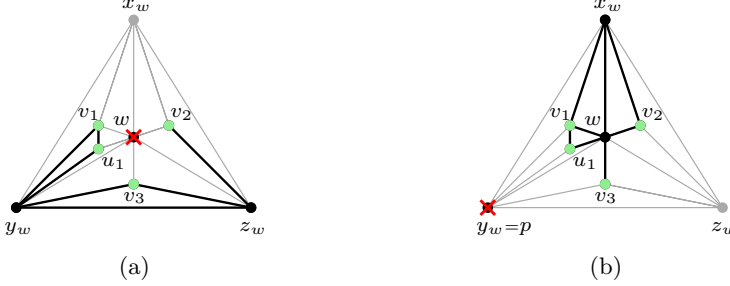


Figure 9.4: Illustrations for Case 3.3.

paths between y_w and z_w . Thus, $G[\mathcal{I}]$ is outerplanar. In the second subcase, we have that $z_w \notin \mathcal{I}'$ (the argument is analogous when $y_w \notin \mathcal{I}'$). Indeed, the cycle $x_w w y_w$ in $G[\mathcal{I}']$ is replaced by the cycle $x_w v_1 y_w$. Additionally, u_1 has degree 2 in $G[\mathcal{I}]$, and any path connecting x_w and y_w that passes through u_1 must go through v_1 . Overall, this does not increase the number of internally disjoint paths between x_w and y_w . Also, the two sets of vertices $\{v_2, u_2\}$ and $\{v_3\}$ form two biconnected components in $G[\mathcal{I}]$ with x_w and y_w , respectively (the first is either a cycle of length 3 or a path of length 2 depending on the neighbors of u_2 , while the second is a path of length 1), such that x_w and y_w are cutvertices. Overall, this shows that $G[\mathcal{I}]$ is outerplanar. Moreover, in both subcases, since $|V(G)| = |V(G')| + 5$ and $|\mathcal{I}| \geq |\mathcal{I}'| + 4$, it follows that $\frac{|\mathcal{I}|}{|V(G)|} \geq \frac{2}{3}$. Note that, in the first subcase, the specific neighbors of u_1 and u_2 are not relevant to the argument, while in the second subcase, only those of u_2 are not relevant.

Case 3.3: $\langle 2, 1, 1 \rangle$. In this case, w has three children v_1, v_2 , and v_3 in T_G , where v_1 has a child u_1 . We apply the induction on the graph $G' := G[(V \setminus T_G(w)) \cup \{w\}]$. Let $\mathcal{I}'$ be the vertex set obtained by induction such that $G[\mathcal{I}']$ is outerplanar. Without loss of generality, we may assume that $w \in \mathcal{I}'$ and that one of x_w, y_w , and z_w is not in $\mathcal{I}'$. We distinguish two subcases, depending on which of x_w or z_w is not in $\mathcal{I}'$ (the case where y_w is not in $\mathcal{I}'$ is symmetric to the latter). In the former subcase (see Figure 9.4a), we define $\mathcal{I} := (\mathcal{I}' \setminus \{w\}) \cup \{v_1, v_2, v_3, u_1\}$. In the latter subcase, we define $\mathcal{I} := (\mathcal{I}' \setminus \{p\}) \cup \{v_1, v_2, v_3, u_1\}$, where $p = x_w$ if u_1 is a neighbor of x_w , and $p = y_w$ otherwise (see Figure 9.4b when $p = y_w$). We prove that $G[\mathcal{I}]$ is outerplanar. Indeed, in the first subcase, the cycle $y_w w z_w$ in $G[\mathcal{I}']$ is replaced by the cycle $y_w v_3 z_w$, which does not increase the number of internally disjoint paths between y_w and z_w . The two sets of vertices $\{v_1, u_1\}$ and $\{v_2\}$ form two

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

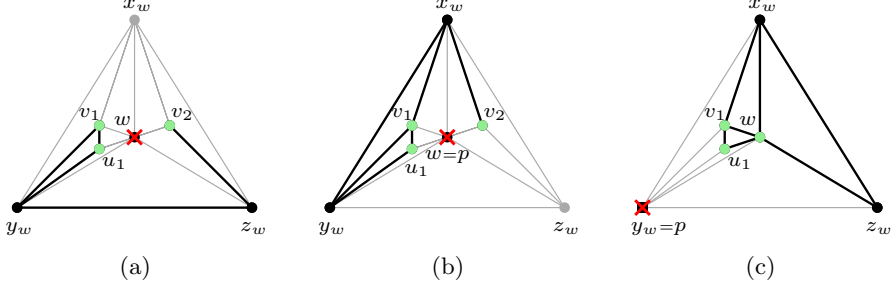


Figure 9.5: Illustrations for (a)-(b) Case 3.4 and (c) Case 3.5.

biconnected components in $G[\mathcal{I}]$ with y_w and z_w , respectively (the first is either a cycle of length 3 or a path of length 2 depending on the neighbors of u_1 , while the second is a path of length 1), such that y_w and z_w are cutvertices. Also, v_3 has only y_w and z_w as neighbors in $G[\mathcal{I}]$. Hence, this does not increase the number of internally disjoint paths between y_w and z_w . Thus, $G[\mathcal{I}]$ is outerplanar. In the second subcase, we have that $z_w \notin \mathcal{I}'$ (the argument is analogous when $y_w \notin \mathcal{I}'$). Indeed, the removal of p results in a biconnected component in $G[\mathcal{I}]$ formed by the set of vertices $\{v_1, v_2, v_3, u_1\}$, which is outerplanar (see Figure 9.4b), and such that the vertex in $\{x_w, y_w\}$ that is not p is a cutvertex. Hence, this shows that $G[\mathcal{I}]$ is outerplanar. Moreover, since $|V(G)| = |V(G')| + 4$ and $|\mathcal{I}| \geq |\mathcal{I}'| + 3$, it follows that $\frac{|\mathcal{I}|}{|V(G)|} \geq \frac{2}{3}$. Note that, in the first subcase, the specific neighbors of u_1 are not relevant to the argument.

Case 3.4: $\langle 2, 1, 0 \rangle$. In this case, w has two children v_1 and v_2 in T_G , where v_1 has a child u_1 . We apply the induction on the graph $G' := G[(V \setminus T_G(w)) \cup \{w\}]$. Let $\mathcal{I}'$ be the vertex set obtained by induction such that $G[\mathcal{I}']$ is outerplanar. Without loss of generality, we may assume that $w \in \mathcal{I}'$ and that one of x_w, y_w , and z_w is not in $\mathcal{I}'$. We distinguish two subcases, depending on which of x_w or z_w is not in $\mathcal{I}'$ (the case where y_w is not in $\mathcal{I}'$ is symmetric to the first). If $x_w \notin \mathcal{I}'$ (see Figure 9.5a), we define $\mathcal{I} := (\mathcal{I}' \setminus \{w\}) \cup \{v_1, v_2, u_1\}$. If $z_w \notin \mathcal{I}'$, we define $\mathcal{I} := (\mathcal{I}' \setminus \{p\}) \cup \{v_1, v_2, u_1\}$, where $p = x_w$ if u_1 is a neighbor of x_w , and $p = w$ otherwise (see Figure 9.5b when $p = w$). We prove that $G[\mathcal{I}]$ is outerplanar. In the first subcase, we have that $x_w \notin \mathcal{I}'$ (the argument is analogous when $y_w \notin \mathcal{I}'$). Indeed, the two sets of vertices $\{v_1, u_1\}$ and $\{v_2\}$ form two biconnected components in $G[\mathcal{I}]$ with y_w and z_w , respectively (the first is either a cycle of length 3 or a path of length 2 depending on the neighbors of u_1 , while the second is a path of length 1), such that y_w and z_w are cutvertices.

9. LARGE INDUCED OUTERPLANAR SUBGRAPHS IN PLANAR 3-TREES

Hence, this does not increase the number of internally disjoint paths between y_w and z_w . Thus, $G[\mathcal{I}]$ is outerplanar. In the second subcase, the cycle $x_w w y_w$ in $G[\mathcal{I}']$ is replaced by the cycle $x_w v_1 y_w$ (if $p = w$), or by the cycle $y_w v_1 w$ (if $p = x_w$). Additionally, u_1 has degree 2 in $G[\mathcal{I}]$, and if $p = w$, any path connecting x_w and y_w that passes through u_1 must go through v_1 ; if $p = x_w$, any path connecting y_w and w that passes through u_1 must go through v_1 . Overall, this does not increase the number of internally disjoint paths between the two vertices in $\{x_w, y_w, w\}$ that are in $\mathcal{I}$. Also, vertex v_2 forms a biconnected component in $G[\mathcal{I}]$ either with x_w (if $p = w$), or with w (if $p = x_w$), which is a path of length 1, such that either x_w or w is a cutvertex. Hence, this shows that $G[\mathcal{I}]$ is outerplanar. Moreover, since $|V(G)| = |V(G')| + 3$ and $|\mathcal{I}| \geq |\mathcal{I}'| + 2$, it follows that $\frac{|\mathcal{I}|}{|V(G)|} \geq \frac{2}{3}$. Note that, in the first subcase, the specific neighbors of u_1 are not relevant to the argument.

Case 3.5: $\langle 2, 0, 0 \rangle$. In this case, w has only one child v_1 in T_G , where v_1 has a child u_1 . We apply the induction on the graph $G' := G[(V \setminus T_G(w))]$. Let $\mathcal{I}'$ be the vertex set obtained by induction such that $G[\mathcal{I}']$ is outerplanar. We may assume that x_w, y_w , and z_w are all in $\mathcal{I}'$. We define $\mathcal{I} := (\mathcal{I}' \setminus \{p\}) \cup \{w, v_1, u_1\}$, where $p = x_w$ if u_1 is a neighbor of x_w , and $p = y_w$ otherwise (see [Figure 9.5c](#) when $p = y_w$). We prove that $G[\mathcal{I}]$ is outerplanar. Indeed, the cycle $x_w z_w y_w$ in $G[\mathcal{I}']$ is replaced by the cycle $x_w w z_w$ (if $p = y_w$), or by the cycle $y_w w z_w$ (if $p = x_w$). Additionally, u_1 has degree 2 in $G[\mathcal{I}]$, with one of its two neighbors in $G[\mathcal{I}]$ being v_1 . Moreover, if $p = y_w$, any path connecting x_w and z_w that passes through v_1 or u_1 must go through w ; if $p = x_w$, any path connecting y_w and z_w that passes through v_1 or u_1 must go through w . Overall, this does not increase the number of internally disjoint paths between the two vertices in $\{x_w, y_w, z_w\}$ that are in $\mathcal{I}$. Furthermore, since $|V(G)| = |V(G')| + 3$ and $|\mathcal{I}| \geq |\mathcal{I}'| + 2$, it follows that $\frac{|\mathcal{I}|}{|V(G)|} \geq \frac{2}{3}$. Note that, in this case, the specific neighbors of u_1 are not relevant to the argument.

This completes the discussion of Case 3, and we can therefore assume that T_G contains no vertex w having a child v with exactly one child, which is a leaf. This concludes the proof. $\square$

Chapter 10

Conclusions and Open Problems

The second part of this thesis addressed [Question 0.3](#) by investigating the size of large constrained induced subgraphs in planar graphs. For all the graph families considered, we established both lower and upper bounds on the value of the corresponding function, namely $f(n)$ or $f_k(n)$, depending on the imposed constraints. We remark that our upper bounds are existential, whereas our lower bounds are universal and are based on polynomial-time algorithms that compute the desired subgraphs. We now summarize the main contributions and open problems of each chapter.

Induced subgraphs of bounded degree in outerplanar and planar graphs. In [Chapter 7](#), we initiated the study of large induced graphs of bounded degree in classes of planar graphs. For any integer $k \geq 3$, we presented upper and lower bounds for the value of the function $f_k(n)$ such that every n -vertex outerplanar or planar graph has an induced subgraph with degree at most k and with $f_k(n)$ vertices. An obvious direction for further research is to close the gaps between the upper and lower bounds we presented. See again [Table 7.1](#) in [Section 7.1](#) for a numerical comparison of our upper and lower bounds for the case $k = 3$.

We believe that our techniques have the potential to yield further improvements on the considered problems, as explained below.

Let c_k be the maximum value such that every n -vertex outerplanar graph G has an induced subgraph with degree at most k and with $c_k \cdot n + o(n)$ vertices. We believe that the use of [Lemmas 7.3](#) and [7.4](#) that was done in order to prove [Theorems 7.5](#) and [7.6](#) could be stressed further, in order to prove a lower bound

10. CONCLUSIONS AND OPEN PROBLEMS

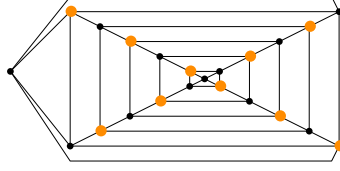


Figure 10.1: An n -vertex 4-regular planar graph in which repeatedly removing degree-4 vertices might result in the removal of $\frac{n}{2} - \mathcal{O}(1)$ vertices. The removed vertices are large disks.

of the form $d_k \cdot n + \mathcal{O}(1)$ for the size of a set $\mathcal{I}$ such that $G[\mathcal{I}]$ has degree at most k , where d_k is arbitrarily close to c_k . Indeed, by employing “very large” values of ℓ in [Lemma 7.3](#), the numerical effect of the constraint imposed by [Lemma 7.4](#) of excluding two vertices, namely u and v , from the set of vertices $\mathcal{I}_H$ that induce a large subgraph $G[\mathcal{I}_H]$ with degree at most k in the uv -split subgraph H of G becomes negligible. We plan to use this intuition to experimentally determine the value c_k , for small values of k , up to some decimal digits. In particular, it would be interesting to understand whether c_k is equal to $\frac{k-1}{k}$, which would match our upper bound from [Theorem 7.1](#). [Theorem 7.5](#) proves that $c_k = \frac{k-1}{k}$ is indeed the right bound for $k = 3$. Also, we suspect that these techniques could be extended to find large induced subgraphs of bounded degree in graphs of bounded treewidth.

In order to find large induced subgraphs of degree at most k in a planar graph G , we employed a greedy algorithm that repeatedly removes a vertex with maximum degree, until the graph has degree at most k . For $k = 3$ and $k = 4$, we modified the algorithm so that the vertex removal stops when G has degree at most $k + 1$. Then G is augmented so that it becomes $(k + 1)$ -regular. Finally, G achieves degree at most k by the removal of a small dominating set, for which we have good upper bounds on the minimum size [\[217, 218\]](#).

It is worth remarking that the greedy algorithm without the dominating set strategy still produces a lower bound better than $\lceil \frac{n}{3} \rceil$, which is the size of a linear forest that one is guaranteed to find in G [\[193\]](#). However, for the case in which $d = 4$ and $k = 3$, it only gives us an $\frac{n}{2}$ lower bound, see [Figure 10.1](#), which is worse than the $\frac{7n}{11}$ lower bound we get by using the dominating set strategy, see [Theorem 7.8](#); consequently, not employing the dominating set strategy leads to worse bounds also for $d = 5$ and $d \geq 6$.

We also ask whether one could obtain lower bounds better than ours for $k \geq 5$ by using known results on the size of dominating sets in $(k + 1)$ -regular

graphs [223].

Observe that the degree of a planar graph can be reduced to k by repeatedly applying the dominating set strategy, each time decreasing the degree of the current graph by 1. This approach does not seem to lead to good bounds for our problem.

We remark that our greedy algorithm does not exploit the planarity of the graph, other than for bounding its number of edges. Hence, we find interesting to understand whether planarity can be exploited more deeply to get better bounds. In particular, lower bounds better than the ones presented in this chapter could be obtained if one could prove that every planar graph with maximum degree 4 or with maximum degree 5 has a subset of vertices whose size is smaller than $\frac{4n}{11}$ or $\frac{5n}{14}$, respectively, such that every degree-4 vertex is in the set or adjacent to a vertex in the set.

Induced outerplanar subgraphs in 2-outerplanar graphs. In Chapter 8, we focused on large induced outerplanar subgraphs in 2-outerplanar graphs. In particular, we recovered a result of [15] that shows a tight bound for the value of the function $f(n)$ such that every n -vertex 2-outerplane graph has an induced outerplane subgraph with $f(n)$ vertices. Our result relies on a different, and unfortunately much more complex, proof.

We point out some open problems, all of which are related to the Albertson and Berman's conjecture. First, as mentioned in Chapter 6, one could approach the conjecture by identifying a large outerplanar induced subgraph in any n -vertex planar graph and then exploiting the fact that outerplanar graphs always contain induced forests with at least $\frac{2}{3}n$ vertices. In particular, improving the guaranteed size of such outerplanar induced subgraphs beyond a $\frac{3}{5}$ fraction of the vertices would immediately yield a stronger bound toward the conjecture. Second, 3-outerplanar graphs constitute the next natural class to investigate. We ask whether any 3-outerplanar graph on n vertices contains an induced outerplane graph of size at least $\frac{2}{3}n$ [15, Conjecture 1], and whether any 3-outerplanar graph on n vertices contains an induced forest of size at least $\frac{n}{2}$. In the former case, by the result of [188], this would imply a better lower bound toward the conjecture for the class of 3-outerplanar graphs, while in the latter case it would prove the conjecture for the same class. Finally, we ask whether any n -vertex planar graph contains an induced outerplane graph of size at least $\frac{2}{3}n$ [15, Conjecture 2], which would improve the known lower bound for the conjecture.

Induced outerplanar subgraphs in planar 3-trees. In Chapter 9, we investigated structural bounds on the size of large induced outerplanar subgraphs

10. CONCLUSIONS AND OPEN PROBLEMS

in planar 3-trees. In particular, we established both upper and lower bounds for the value of the function $f(n)$ such that every n -vertex planar 3-tree has an induced outerplanar subgraph with $f(n)$ vertices. For the upper bound, we presented a family of planar 3-trees that attains the bound for any value of n . For the lower bound, we provided an inductive proof that exploits the inherent tree-like structure of planar 3-trees. We believe that the inductive argument can be strengthened to show that the same lower bound holds even when the input planar 3-tree is given together with a fixed planar embedding, and the induced subgraph is required to be outerplane with respect to this embedding, similarly to the main result of [Chapter 8](#). Moreover, the construction used to establish the upper bound could potentially be extended to derive analogous upper bounds for broader classes of graphs with larger treewidth. Finally, we pose the question of whether the tight bound on the size of the largest induced outerplanar subgraph in an n -vertex planar 3-tree is $\frac{7}{10}n$. Answering this question likely requires a fundamentally different proof technique, as a direct inductive approach with case analysis would rely on a deep complex analysis.

Appendices

Appendix A

Other Research Activities

A.1 Upward Pointset Embeddings of Planar st -Graphs

A.1.1 Introduction

Given an n -vertex upward planar graph G and a set S of n points in the plane, an *upward pointset embedding* (UPSE) of G on S is an upward planar drawing of G where the vertices are mapped to the points of S and the edges are represented as straight-line segments. The UPWARD POINTSET EMBEDDABILITY TESTING PROBLEM (UPSE TESTING) asks whether an upward planar graph G has an UPSE on a given pointset S .

Pointset embedding problems are classic challenges in Graph Drawing and have been considered for both undirected and directed graphs. For an undirected graph, a *pointset embedding* (PSE) has the same definition of an UPSE, except that the drawing must be planar, rather than upward planar. The POINTSET EMBEDDABILITY TESTING PROBLEM (PSE TESTING) asks whether a planar graph has a PSE on a given pointset S . Pointset embeddings have been studied by several authors. It is known that a graph admits a PSE on *every* pointset in general position if and only if it is outerplanar [224, 225]; such a PSE can be constructed efficiently [226, 227, 228, 229]. PSE TESTING is, in general, NP-complete [230], however it is polynomial-time solvable if the input graph is a planar 3-tree [231, 232]. More in general, a polynomial-time algorithm for PSE TESTING exists if the input graph has a fixed embedding, bounded treewidth, and bounded face size [233]. PSE becomes NP-complete if one of the latter two conditions does not hold. PSEs have been studied also for dynamic

graphs [234].

The literature on UPSEs is not any less rich than the one on PSEs. From a combinatorial perspective, the directed graphs with an UPSE on a one-sided convex pointset have been characterized [235, 236]; all directed trees are among them. Conversely, there exist directed trees that admit no UPSE on certain convex pointsets [235]. Directed graphs that admit an UPSE on any convex pointset, but not on any pointset in general position, exist [237]. It is still unknown whether every digraph whose underlying graph is a path admits an UPSE on every pointset in general position, see, e.g., [238]. UPSEs where bends along the edges are allowed have been studied in [235, 239, 240, 241, 242]. From the computational complexity point of view [243, 244], it is known that UPSE TESTING is NP-hard, even for planar *st*-graphs and 2-convex pointsets, and that UPSE TESTING can be solved in polynomial time if the given pointset is convex.

Chapter contributions. In this chapter¹, we tackle UPSE TESTING for planar *st*-graphs. Let G be an n -vertex planar *st*-graph and S be a set of n points in the plane. We adopt the common assumption in the context of upward pointset embeddability, see e.g. [237, 235, 243, 244], that no two points of S lie on the same horizontal line. Our results are the following:

- In Section A.1.3, we show that UPSE TESTING is NP-hard even if G consists of a set of internally-disjoint *st*-paths (Theorem A.1.1). A similar proof shows that UPSE TESTING is NP-hard for directed trees consisting of a set of directed root-to-leaf paths (Theorem A.1.2). This answers an open question from [247] and strengthens a result therein, which shows NP-hardness for directed trees with multiple sources and with a prescribed mapping for a vertex.
- In Section A.1.4, we show that UPSE TESTING can be solved in $\mathcal{O}(n^{4k})$ time and $\mathcal{O}(n^{3k})$ space, where k is the size of the largest *st*-cutset of G (Theorem A.1.7). This parameter measures the “fatness” of the digraph and coincides with the length of the longest directed path in the dual [248]. By leveraging on the techniques developed for the testing algorithm, we also show how to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time (Theorem A.1.8).

¹The content of this chapter appeared in [245, 246] and is joint work with Carlos Alegría-Galicia, Susanna Caroppo, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, and Maurizio Patrignani.

- In [Section A.1.5](#), we provide a simple characterization of the pointsets in general position that allow for an UPSE of G , if G consists of two (internally-disjoint) st -paths. Based on that, we provide an $\mathcal{O}(n \log n)$ testing algorithm for this case ([Theorem A.1.9](#)).
- Finally, in [Section A.1.6](#), inspired by the fact that an UPSE of a planar st -graph composed of two st -paths defines a non-crossing monotone Hamiltonian cycle on S , we provide an algorithm that enumerates all the non-crossing monotone Hamiltonian cycles on a given pointset with $\mathcal{O}(n)$ worst-case delay, and $\mathcal{O}(n^2)$ space usage and set-up time ([Theorem A.1.10](#)).

Concerning our last result, we remark that a large body of research has considered problems related to enumerating and counting non-crossing structures on a given pointset [[249](#), [250](#), [251](#), [252](#), [253](#)]. Despite this effort, the complexity of counting the non-crossing Hamiltonian cycles, often called *polygonalizations*, remains open [[254](#), [252](#), [255](#)]. However, it is possible to enumerate all polygonalizations of a given pointset in singly-exponential time [[256](#), [257](#)]. Recently, an algorithm has been shown [[258](#)] to enumerate all polygonalizations of a given pointset in time polynomial in the output size, i.e., bounded by a polynomial in the number of solutions. However, an enumeration algorithm with polynomial (in the input size) delay is not yet known, neither in the worst-case nor in the average-case acceptance. Our enumeration algorithm achieves this goal for the case of monotone polygonalizations.

A.1.2 Preliminaries

We use standard terminology in graph theory [[25](#)] and graph drawing [[14](#)]. For an integer $k > 0$, let $[k]$ denote the set $\{1, \dots, k\}$. A *permutation with repetitions* of k elements from U is an arrangement of any k elements of a set U , where repetitions are allowed.

For a point $p \in \mathbb{R}^2$, we denote by $x(p)$ and $y(p)$ the x - and y -coordinate of p , respectively. The *convex hull* $\mathcal{CH}(S)$ of a set S of points in $\mathbb{R}^2$ is the union of all convex combinations of points in S . The *boundary* $\mathcal{B}(S)$ of $\mathcal{CH}(S)$ is the polygon with minimum perimeter enclosing S . The points of S with lowest and highest y -coordinates are the *south* and *north extreme* of S , respectively; we also refer to them as to the *extremes* of S . The *left envelope* of S is the subpath $\mathcal{E}_L(S)$ of $\mathcal{B}(S)$ that lies to the left of the line passing through the extremes of S ; it includes the extremes of S . The *right envelope* $\mathcal{E}_R(S)$ of S

is defined analogously. We denote the subset of S in $\mathcal{E}_L(S)$ and in $\mathcal{E}_R(S)$ by $\mathcal{H}_L(S)$ and $\mathcal{H}_R(S)$, respectively. A *polyline* $(p_1, \dots, p_k)$, with $k \geq 2$, is a chain of straight-line segments.

We call *ray* any of the two half-lines obtained by cutting a straight line at any of its points, which is the *starting point* of the ray. A ray is *upward* if it passes through points whose y -coordinate is larger than the one of the starting point of the ray. We denote by $\rho(p, q)$ the ray starting at a point p and passing through a point q . For a set of points S and a point p whose y -coordinate is smaller than the one of every point in S , we denote by $\rho(p, S)$ (by $\ell(p, S)$) the rightmost (resp. leftmost) upward ray starting at p and passing through a point of S , that is, the clockwise (resp. counter-clockwise) rotation around p which brings $\rho(p, S)$ (resp. $\ell(p, S)$) to coincide with any other upward ray starting at p and passing through a point of S is larger than 180° .

A polyline $(p_1, \dots, p_k)$ is *y-monotone* if $y(p_i) < y(p_{i+1})$, for $i = 1, \dots, k-1$. A *monotone path* on a pointset S is a y -monotone polyline $(p_1, \dots, p_k)$ such that the points $p_1, \dots, p_k$ belong to S . A *monotone cycle* on S consists of two monotone paths on S that share their endpoints. A *monotone Hamiltonian cycle* $(p_1, \dots, p_k, p_1)$ on S is a monotone cycle on S such that each point of S is a point p_i (and vice versa).

A path $(v_1, \dots, v_k)$ is *directed* if, for $i = 1, \dots, k-1$, the edge (v_i, v_{i+1}) is directed from v_i to v_{i+1} ; the vertices $v_2, \dots, v_{k-1}$ are *internal*. A *planar st-graph* is an acyclic digraph with one source s and one sink t , which admits a planar embedding in which s and t are on the boundary of the outer face. An *st-path* in a planar *st-graph* is a directed path from s to t . A drawing of a directed graph is *straight-line* if each edge is represented by a straight-line segment, it is *planar* if no two edges cross, and it is *upward* if every edge is represented by a Jordan arc monotonically increasing in the y -direction from the tail to the head. A digraph that admits an upward planar drawing is an *upward planar graph*. Every upward planar graph admits an upward planar straight-line drawing [248]. An *Upward Pointset Embedding* (*UPSE*, for short) of an upward planar graph G on a pointset S is an upward planar straight-line drawing of G that maps each vertex of G to a point in S . In this chapter, we study the following problem.

UPWARD POINTSET EMBEDDABILITY TESTING PROBLEM (UPSE TESTING)

Input: An n -vertex upward planar graph G and a pointset $S \subset \mathbb{R}^2$ with $|S| = n$.
Question: Does there exist an UPSE of G on S ?

In the remainder, we assume that not all points in S lie on the same line, as otherwise there is an UPSE if and only if the input is a directed path. Recall that no two points in S have the same y -coordinate. Unless otherwise specified, we do not require points to be in *general position*, i.e., we allow three or more points to lie on the same line.

A.1.3 NP-Completeness of UPSE Testing

In this section we prove that UPSE TESTING is NP-complete. The membership in NP is obvious, as one can non-deterministically assign the vertices of the input graph G to the points of the input pointset S and then test in polynomial time whether the assignment results in an upward planar straight-line drawing of G . In the remainder of the section, we prove that UPSE TESTING is NP-hard even in very restricted cases.

We first show a reduction from 3-PARTITION to instances of UPSE in which the input is a planar st -graph composed of a set of internally-disjoint st -paths. An instance of 3-PARTITION consists of a set $A = \{a_1, \dots, a_{3b}\}$ of $3b$ integers, where $\sum_{i=1}^{3b} a_i = bB$ and $B/4 \leq a_i \leq B/2$, for $i = 1, \dots, 3b$. The 3-PARTITION problem asks whether A can be partitioned into b subsets $A_1, \dots, A_b$, each with three integers, so that the sum of the integers in each set A_i is B . Since 3-PARTITION is strongly NP-hard [259], we may assume that B is bounded by a polynomial function of b . Given an instance A of 3-PARTITION, we show how to construct in polynomial time, precisely $\mathcal{O}(b \cdot B)$, an equivalent instance (G, S) of UPSE TESTING.

The n -vertex planar st -graph G is composed of $4b + 1$ internally-disjoint st -paths. Namely, for $i = 1, \dots, 3b$, we have that G contains an a_i -path, i.e., a path with a_i internal vertices, and $b + 1$ additional k -paths, where $k = 2B + 1$. Note that $n = 2 + (b + 1)k + \sum_{i=1}^{3b} a_i = 2 + (b + 1)k + bB$.

The points of S lie on the plane as follows (see Figure A.1.1a):

- p_1 is the origin, with coordinates $(0, 0)$.
- Consider $b + 1$ upward rays $\rho_1, \dots, \rho_{b+1}$, whose starting point is p_1 , such that the angles $\alpha_1, \dots, \alpha_{b+1}$ that they respectively form with the x -axis satisfy $3\pi/4 > \alpha_1 > \dots > \alpha_{b+1} > \pi/4$. Let ℓ be a line intersecting all the rays, with a positive slope smaller than $\pi/4$. For $j = 1, \dots, b + 1$, place k points $p_{j,1}, \dots, p_{j,k}$ (in this order from bottom to top) along ρ_j , so that $p_{j,k}$ is on ℓ and no two points share the same y -coordinate. Observe that $p_{b+1,k}$ is the highest point placed so far.

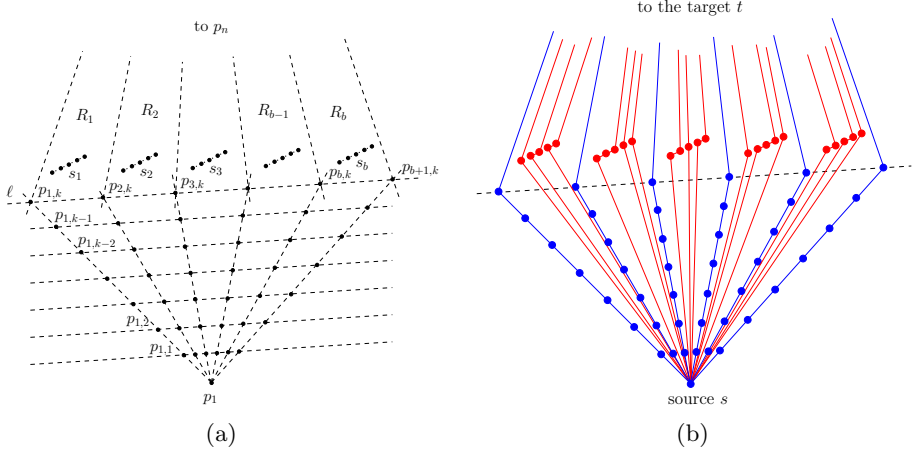


Figure A.1.1: Illustration for the proof of [Theorem A.1.1](#). (a) The pointset S . (b) The UPSE of G on S .

- Place p_n at coordinates $(0, 10 \cdot y(p_{b+1,k}))$.
- Finally, for $j = 1, \dots, b$, place B points along a non-horizontal segment s_j in such a way that: (i) s_j is entirely contained in the triangle with vertices $p_{j,k}$, $p_{j+1,k}$, and p_n , (ii) for any point p on s_j , the polygonal line $\overline{p_1 p} \cup \overline{p p_n}$ is contained in the region R_j delimited by the polygon $\overline{p_1 p_{j,k}} \cup \overline{p_{j,k} p_n} \cup \overline{p_n p_{j+1,k}} \cup \overline{p_{j+1,k} p_1}$, and (iii) no two distinct points on any two segments s_i and s_j share the same y -coordinate.

Note that S has $2 + (b+1)k + bB = n$ points. The described reduction is the main ingredient for the proof of the following theorem.

Theorem A.1.1. UPSE TESTING is NP-hard even for planar st -graphs consisting of a set of directed internally-disjoint st -paths.

Proof. First, the construction of G and S takes polynomial time. In particular, the coordinates of the points in S can be encoded with a polylogarithmic number of bits. In order to prove the NP-hardness, it remains to show that the constructed instance (G, S) of UPSE TESTING is equivalent to the given instance A of 3-PARTITION. Refer to [Figure A.1.1b](#).

Suppose first that A is a positive instance of 3-PARTITION, that is, there exist sets $A_1, \dots, A_b$, each with three integers, such that the sum of the integers

in each set A_j is B . We construct an UPSE of G on S as follows. We map s to p_1 and t to p_n . For $j = 1, \dots, b+1$, we map the k internal vertices of a k -path to the points $p_{j,1}, \dots, p_{j,k}$, so that vertices that come first in the directed path have smaller y -coordinates. Furthermore, for $j = 1, \dots, b$, let $A_j = \{a_{j_1}, a_{j_2}, a_{j_3}\}$. Then we map the a_{j_1} internal vertices of an a_{j_1} -path, the a_{j_2} internal vertices of an a_{j_2} -path, and the a_{j_3} internal vertices of an a_{j_3} -path to the set of B points in the triangle with vertices $p_{j,k}$, $p_{j+1,k}$, and p_n , so that vertices that come first in the directed paths have smaller y -coordinates and so that the internal vertices of the a_{j_1} -path have smaller y -coordinates than the internal vertices of the a_{j_2} -path, which have smaller y -coordinates than the internal vertices of the a_{j_3} -path. This results in an UPSE of G on S .

Suppose next that (G, S) is a positive instance of UPSE TESTING. Trivially, in any UPSE of G on S , we have that s is drawn on p_1 and t on p_n . Consider the points $p_{1,1}, \dots, p_{b+1,1}$. The paths using them use all the $(b+1)k$ points $p_{j,i}$, with $j = 1, \dots, b+1$ and $i = 1, \dots, k$. Indeed, if these paths left one of such points unused, no other path could reach it from s without passing through $p_{1,1}, \dots, p_{b+1,1}$, because of the collinearity of the points along the rays $\rho_1, \dots, \rho_{b+1}$. Hence, there are at most $b+1$ paths that use $(b+1)k$ points. Since the a_i -paths have less than k internal vertices, these $b+1$ paths must all be k -paths. Let $P_1, \dots, P_{b+1}$ be the left-to-right order of the k -paths around p_1 . For $j = 1, \dots, b+1$, path P_j uses all points $p_{j,i}$ on ρ_j , as if P_j used a point $p_{h,i}$ with $h > j$, then two among $P_j, \dots, P_{b+1}$ would cross each other. Note that, after using $p_{j,k}$, path P_j ends with the segment $\overline{p_{j,k}p_n}$. Hence, for $j = 1, \dots, b$, the region R_j is bounded by P_j and P_{j+1} ; recall that R_j contains the segment s_j . The a_i -paths must then use the points on $s_1, \dots, s_b$. Since $B/4 < a_i < B/2$, no two a_i -paths can use all the B points in one region and no four a_i -paths can lie in the same region. Hence, three a_i -paths use the B points in each region, and this provides a solution to the given 3-PARTITION instance. $\square$

We next reduce the 3-PARTITION problem to the instances of UPSE TESTING in which the input is a directed tree consisting of a set of root-to-leaf paths. Consider an instance of 3-PARTITION consisting of a set $A = \{a_1, \dots, a_{3b}\}$ of $3b$ integers, where $\sum_{i=1}^{3b} a_i = bB$ and $B/4 \leq a_i \leq B/2$, for $i = 1, \dots, 3b$. We construct a directed tree T as follows. The root s of T has $4b-1$ children. Among them, $b-1$ are leaves $v_1, \dots, v_{b-1}$, while each of the remaining $3b$ children is the first vertex of a directed path P_i , for $i = 1, \dots, 3b$, consisting of the $a_i + 1$ vertices $v_{i,1}, v_{i,2}, \dots, v_{i,a_i+1}$, where $v_{i,1}$ is the child of s and v_{i,a_i+1} is a leaf. All the edges of T are directed from the root s to the leaves. Note

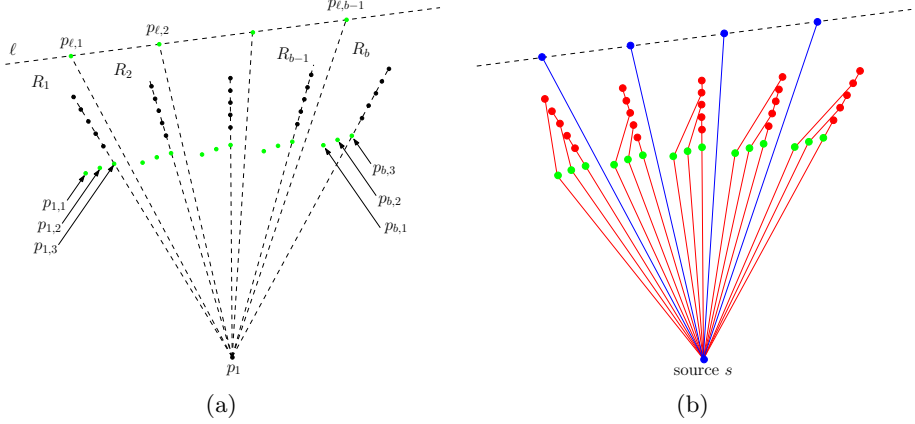


Figure A.1.2: Illustration for the proof of [Theorem A.1.2](#). (a) The pointset S . The points of S visible from p_1 (green points) are as many as the children of the root of the tree T . (b) The UPSE of T on S corresponding to a solution to the original instance 3-PARTITION (red vertices).

that the number of vertices of T is $n = 1 + (b - 1) + \sum_{i=1}^{3b} (a_i + 1) = b(B + 4)$. The points of S lie on the plane as follows (see [Figure A.1.2a](#)):

- p_1 is the origin, with coordinates $(0, 0)$.
- Consider $b - 1$ upward rays $\rho_1, \dots, \rho_{b-1}$, whose starting point is p_1 , such that the angles $\alpha_1, \dots, \alpha_{b-1}$ formed by $\rho_1, \dots, \rho_{b-1}$ with the x -axis satisfy $3\pi/4 > \alpha_1 > \dots > \alpha_{b-1} > \pi/4$. These rays split the half plane above the x -axis into b regions R_j , with $j = 1, 2, \dots, b$. In the interior of each region R_j , place three points $p_{j,1}, p_{j,2}$, and $p_{j,3}$ in such a way that $p_{j,1}$ is lower than $p_{j,2}$, which is lower than $p_{j,3}$, and so that they are all visible from s . Along the line passing through s and $p_{j,3}$ place B points above $p_{j,3}$.
- Let y_m be the highest y -coordinate used so far. Let ℓ be a line with positive slope smaller than $\pi/4$ intersecting all the rays $\rho_1, \dots, \rho_{b-1}$ at points that have y -coordinates larger than y_m . For $j = 1, \dots, b - 1$, place a point $p_{\ell,j}$ at the intersection of ρ_j with ℓ .

Note that S has $1 + 3b + bB + (b - 1) = b(B + 4) = n$ points. The described reduction is the main ingredient for the proof of the following theorem.

Theorem A.1.2. UPSE TESTING is NP-hard even for directed trees consisting of a set of directed root-to-leaf paths.

Proof. First, the construction of T and S takes polynomial time. In particular, the coordinates of the points in S can be encoded with a polylogarithmic number of bits. In order to prove the NP-hardness, it remains to show that the constructed instance (T, S) of UPSE TESTING is equivalent to the given instance A of 3-PARTITION. Refer to Figure A.1.2b.

Suppose first that A is a positive instance of 3-PARTITION, that is, there exist sets $A_1, \dots, A_b$, each with three integers, such that the sum of the integers in each set A_j is B . We construct an UPSE of G on S as follows. We map s to p_1 . For $j = 1, \dots, b-1$, we map the child v_j of s to $p_{\ell,j}$. Furthermore, for $j = 1, \dots, b$, let $A_j = \{a_{j_1}, a_{j_2}, a_{j_3}\}$. Then we map the a_{j_1} internal vertices of an a_{j_1} -path, the a_{j_2} internal vertices of an a_{j_2} -path, and the a_{j_3} internal vertices of an a_{j_3} -path to the set of B points in the region R_j , so that the neighbors of s in the a_{j_1} -path, in the a_{j_2} -path, and in the a_{j_3} -path lie on $p_{j,1}$, $p_{j,2}$, and $p_{j,3}$, respectively, so that vertices that come first in the directed paths have smaller y -coordinates, and so that the internal vertices of the a_{j_1} -path have larger y -coordinates than the internal vertices of the a_{j_2} -path, which have larger y -coordinates than the internal vertices of the a_{j_3} -path. This results in an UPSE of T on S .

Suppose next that (T, S) is a positive instance of UPSE TESTING. It is obvious that the root s of T has to be placed on p_1 . From the root s only $4b-1$ points are visible. These are the points $p_{\ell,j}$, for $j = 1, \dots, b-1$, and the points $p_{h,1}, p_{h,2}$, and $p_{h,3}$, for $h = 1, \dots, b$ (all these points are filled green in Figure A.1.2a). Since T has $4b-1$ children, each child must use one of the above points. Consider point $p_{\ell,b-1}$. Since this is the highest point in the set S , the child that uses it must be a leaf. This also holds for $p_{\ell,b-2}$, which is the highest of the remaining points. Iterating this argument we have that the points $p_{\ell,j}$, with $j = 1, \dots, b-1$, must be used by the $b-1$ children of s which are leaves of T . Since all other vertices have smaller y -coordinates, each path P_i , with $i = 1, \dots, 3m$, is constrained to be into a region R_j , with $j = 1, \dots, b$ (see Figure A.1.2b). Since each region R_j contains exactly three points $p_{j,1}, p_{j,2}$, and $p_{j,3}$ visible from s , each region hosts exactly three such paths, which use the remaining B points, and this provides a solution to the given 3-PARTITION instance. $\square$

A.1.4 Testing and Enumeration Algorithms for Planar *st*-Graphs with Maximum *st*-Cutset of Bounded Size

An *st-cutset* of a planar *st*-graph $G = (V, E)$ is a subset W of E such that:

- removing W from E results in a graph consisting of exactly two connected components C_s and C_t ,
- s belongs to C_s and t belongs to C_t , and
- any edge in W has its tail in C_s and its head in C_t .

In this section, we consider instances (G, S) where G is a planar *st*-graph, whose maximum *st*-cutset has bounded size k . In [Theorem A.1.7](#), we show that UPSE TESTING can be solved in polynomial time for such instances (G, S) . Moreover, in [Theorem A.1.8](#), we show how to enumerate all UPSEs of (G, S) with linear delay. The algorithm for [Theorem A.1.7](#) is based on a dynamic programming approach. It exploits the property that, for an *st*-cutset W defining the connected components C_s and C_t , the extensibility of an UPSE Γ' of $C_s \cup W$ on a subset S' of S to an UPSE of G on S only depends on the drawing of the edges of W , and not on the embedding of the remaining vertices of C_s , provided that in Γ' there exists an horizontal line that crosses all the edges of W . The algorithm for [Theorem A.1.8](#) leverages a variation of the dynamic programming table computed by the former algorithm to efficiently test the extensibility of an UPSE of $C_s \cup W$ (in which there exists a horizontal line that crosses all the edges of W) on a subset S' of S to an UPSE of G on S .

The proofs of [Theorems A.1.7](#) and [A.1.8](#) exploit two dynamic programming tables T and Q defined as follows. Each entry of T and Q is indexed by a *key* that consists of a set of $h \leq k$ triplets $\langle e_i, p_i, q_i \rangle$, where, for any $i = 1, \dots, h$, it holds that $e_i \in E(G)$, $p_i, q_i \in S$, and $y(p_i) < y(q_i)$. Moreover, each key $\chi = \bigcup_{i=1}^h \langle e_i, p_i, q_i \rangle$ satisfies the following constraints:

- the set $E(\chi) = \bigcup_{i=1}^h e_i$ is an *st*-cutset of G and, for every i, j , with $i \neq j$, it holds true that $e_i \neq e_j$ (that is, $|E(\chi)| = h$);
- for every i, j , with $i \neq j$, it holds true that $p_i = p_j$ (resp. that $q_i = q_j$) if and only if e_i and e_j have the same tail (resp. the same head); and
- let ℓ_χ be the horizontal line passing through the tail with largest y -coordinate among the edges in $E(\chi)$, i.e., $\ell_\chi := y = y(p_i)$ s.t. $y(p_j) \leq y(p_i)$

for any $\langle e_j, p_j, q_j \rangle \in \chi$; then ℓ_χ intersects all the segments $\overline{p_j q_j}$, possibly at an endpoint.

For brevity, we sometimes say that the edge e_i has its tail (resp. its head) *mapped by* χ on p_i (resp. on q_i). We also say that e_i is *drawn as in* χ if its drawing is the segment $\overline{p_i q_i}$.

Let $\chi = \bigcup_{i=1}^h \langle e_i, p_i, q_i \rangle$ be a key of T and of Q ; see Figure A.1.3a. Let G_χ be the connected component containing s of the graph obtained from G by removing the edge set $E(\chi)$.

The entry $T[\chi]$ contains a Boolean value such that $T[\chi] = \text{True}$ if and only if there exists an UPSE of $G_\chi^+ = G_\chi \cup E(\chi)$ on some subset $S' \subset S$ with $|S'| = |V(G_\chi^+)|$ such that:

- the lowest point p_s of S belongs to S' and s lies on it, and
- for $i = 1, \dots, h$, the edge e_i is drawn as in χ .

If $T[\chi] = \text{False}$, the entry $Q[\chi]$ contains the empty set $\emptyset$. If $T[\chi] = \text{True}$ and $E(\chi)$ coincides with the set of edges incident to s , then $Q[\chi]$ stores the set $\{\perp\}$. If $T[\chi] = \text{True}$ and $E(\chi)$ does not coincide with the set of edges incident to s , $Q[\chi]$ stores the set Φ of keys with the following properties. Let e_τ be any edge whose tail v_τ has maximum y -coordinate among the edges in $E(\chi)$, i.e., $\langle e_\tau, p_\tau, q_\tau \rangle$ is such that $y(p_\tau) \geq y(p_j)$ for any $\langle e_j, p_j, q_j \rangle \in \chi$. For each $\varphi \in \Phi$, we have that:

- $T[\varphi] = \text{True}$;
- $E(\chi) \cap E(\varphi)$ contains all and only the edges in $E(\chi)$ whose tail is not v_τ , and each edge $e_i \in E(\chi) \cap E(\varphi)$ is drawn in φ as it is drawn in χ ; and
- all the edges in $E(\varphi) \setminus E(\chi)$ have v_τ as their head.

Additionally, we store a list Λ of the keys σ such that $T[\sigma] = \text{True}$ and $E(\sigma)$ is the set of edges incident to t . Note that each edge in $E(\sigma)$ has its head mapped by σ to the point $p_t \in S$ with largest y -coordinate.

We use dynamic programming to compute the entries of T and Q in increasing order of $|V(G_\chi)|$. By the definition of T , we have that G admits an UPSE on S if and only if $\Lambda \neq \emptyset$.

First, we initialize all entries of T to **False** and all entries of Q to $\emptyset$.

If $|V(G_\chi)| = 1$, then G_χ only consists of s . We set $T[\chi] = \text{True}$ and $Q[\chi] = \{\perp\}$ for every key $\chi = \bigcup_{i=1}^h \langle e_i, p_i, q_i \rangle$ such that:

- $e_1, \dots, e_h$ are the edges incident to s ;
- $p_1 = \dots = p_h = p_s$; and
- for every distinct i and j in $\{1, \dots, h\}$, we have that p_s, q_i , and q_j are not aligned.

If $|V(G_\chi)| > 1$, we compute $T[\chi]$ and $Q[\chi]$ as follows, see Figure A.1.3b. If two segments $\overline{p_i q_i}$ and $\overline{p_j q_j}$, with $i \neq j$, cross (that is, they share a point that is internal for at least one of the segments), then we leave $T[\chi]$ and $Q[\chi]$ unchanged; in particular, $T[\chi] = \mathbf{False}$ and $Q[\chi] = \emptyset$. Otherwise, we proceed as follows. Let e_τ be any edge whose tail v_τ has maximum y -coordinate among the edges in $E(\chi)$. Let H^- be the set of edges obtained from $E(\chi)$ by removing all the edges having v_τ as their tail, and let H^+ be the set of edges of G having v_τ as their head. We define the set $H := H^- \cup H^+$. We have the following.

Claim A.1.3. H is an st -cutset of G .

Proof. Recall that, since $E(\chi)$ is an st -cutset, removing the edges of $E(\chi)$ from G yields two connected components C_s and C_t such that s belongs to C_s and t belongs to C_t ; see Figure A.1.4a. Let C'_t be the graph consisting of C_t , the vertex v_τ , and the edges having v_τ as their tail (these are the edges in $E(\chi) \setminus H^-$, which are not part of H). Also, let C'_s be the graph obtained by removing from C_s the vertex v_τ and the edges in H^+ (i.e., these are the edges outgoing from v_τ); see Figure A.1.4b. We have that C'_s and C'_t do not share any vertex,

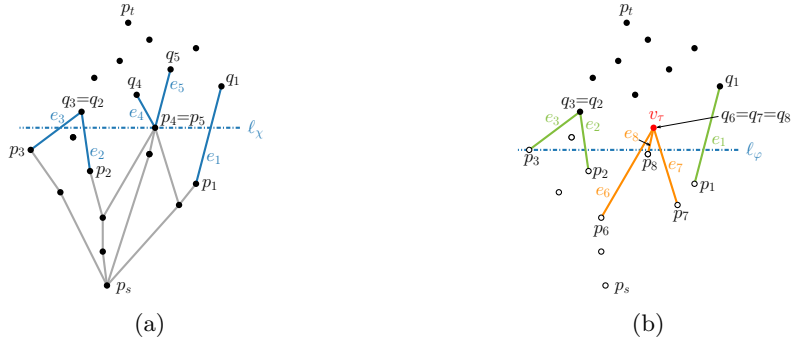


Figure A.1.3: (a) An entry $\chi = \bigcup_{i=1}^5 \langle e_i, p_i, q_i \rangle$ with $T[\chi] = \mathbf{True}$ and a corresponding UPSE of G_χ on a subset of S that includes p_s . (b) An entry φ from which χ stems; the points in $S_\downarrow$ are filled white.

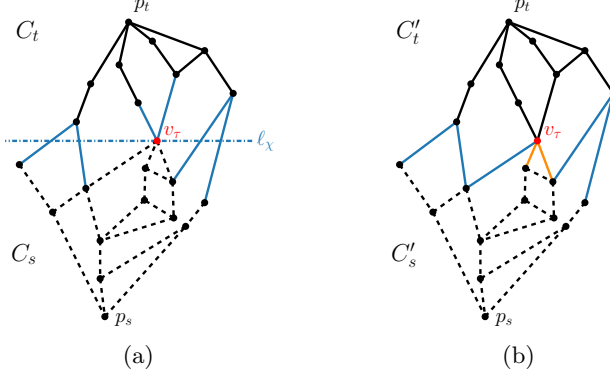


Figure A.1.4: Illustrations for **Claim A.1.3**. (a) The connected components C_s (dashed) and C_t (solid black) defined by the st -cutset $E(\chi)$. (b) The connected components C'_s (dashed) and C'_t (solid black) defined by the st -cutset H (blue and orange edges).

since C_s and C_t do not share any vertex, since $V(C'_s) \subset V(C_s)$ and since the only vertex in $V(C'_t) \setminus V(C_t)$ is v_τ , which does not belong to C'_s . Moreover, by construction $G = C'_t \cup C'_s \cup H$, in particular the only edges connecting vertices in C'_s with vertices in C'_s are those in H , which have their tails in C'_s and their heads in C'_t . Also, we have that s belongs to C'_s and t belongs to C'_t . To prove that H is an st -cutset of G , it only remains to argue that each of C'_s and C'_t is connected. Since $C_t \subseteq C'_t$ and since C_t is connected, we have that every pair of vertices distinct from v_τ is connected by an undirected path in C'_t . Also, the heads of the edges outgoing from v_τ belong to C_t and, by construction, such edges belong to C'_t . Hence, there exists an undirected path in C'_t between v_τ and every vertex of C_t . Therefore, C'_t is connected. Suppose now, for a contradiction, that C'_s is not connected and thus there exists a vertex v which is not in the same connected component as s in C'_s . Since G is a planar st -graph, it contains a directed path from s to v . If such a path does not belong entirely to C'_s , it contains an edge which is directed from a vertex not in C'_s to a vertex in C'_s . Moreover, such an edge belongs to H , however we already observed that all the edges in H are outgoing from the vertices in C'_s , a contradiction. $\square$

Consider the set $S_\downarrow$ consisting of the points in S whose y -coordinates are smaller than $y(p_\tau)$. We have the following crucial observation.

Observation A.1.4. $T[\chi] = \text{True}$ if and only if there exists some key φ , with $E(\varphi) = H$, such that $T[\varphi] = \text{True}$, the edges in H^- are drawn in φ as in χ , the edges in H^+ have their heads mapped by φ on p_τ and their tails on a point in $S_\downarrow$.

In view of **Observation A.1.4**, we can now define a procedure to compute $T[\chi]$ and $Q[\chi]$. Assume that the edges $e_1, \dots, e_{|H^-|}, \dots, e_{|H|} \in H$ are ordered so that the edges of H^- precede those of H^+ . By **Observation A.1.4**, if $|S_\downarrow| < |H^+|$, then we leave $T[\chi]$ and $Q[\chi]$ unchanged, i.e., $T[\chi] = \text{False}$ and $Q[\chi] = \emptyset$. In fact, in this case, there are not enough points in $S_\downarrow$ to map the tails of the edges in H^+ . Otherwise, let D be the set of all permutations with repetitions of $|H^+|$ points from $S_\downarrow$. We define a set Φ of keys that, for each $(d_1, \dots, d_{|H^+|}) \in D$, contains a key φ such that:

- (i) $E(\varphi) = H$;
- (ii) for any $i = 1, \dots, |H^-|$, the triple containing e_i in φ is the same as the triple containing e_i in χ (note that $e_i \in H^-$);
- (iii) for any $j = |H^-| + 1, \dots, |H|$, the triple containing e_j in φ has $q_j = p_\tau$, and $p_j = d_{j-|H^-|}$ (note that $e_j \in H^+$); and
- (iv) for every $i = 1, \dots, |H^-|$ and $j = |H^-| + 1, \dots, |H|$, it holds $p_i = p_j$ if and only if e_i and e_j have the same tail.

Let $\Phi^T = \{\varphi : \varphi \in \Phi \wedge T[\varphi] = \text{True}\}$. By **Observation A.1.4**, we have $T[\chi] = \text{True}$ if and only if $|\Phi^T| \geq 1$. Thus, we set $T[\chi] = \bigvee_{\varphi \in \Phi} T[\varphi]$ and $Q[\chi] = \Phi^T$. We say that χ *stems from* any key $\varphi \in \Phi$ with $T[\varphi] = \text{True}$.

We now upper bound the sizes of T and Q and the time needed to compute them.

Claim A.1.5. *Tables T and Q have size in $\mathcal{O}(n^{3k})$ and $\mathcal{O}(kn^{4k} \log n)$, respectively.*

Proof. First, we give an upper bound on the number of entries of T (and, thus, of Q), which we denote by ρ . Each entry of T is associated with a key χ defined by an *st*-cutset $E(\chi)$ of size at most $h \leq k$, a permutation (possibly with repetitions) of h points in S describing a mapping of the tails of the edges in $E(\chi)$ with points in S , and a permutation (possibly with repetitions) of h points in S describing a mapping of the heads of the edges in $E(\chi)$ with points in S . Recall that $|S| = n$, that $\binom{a}{b} \leq a^b$, and that the number of permutations

with repetition of h elements from a set U is $|U|^h$. Therefore, we have that $\rho \leq \binom{m}{k} \cdot n^k \cdot n^k \leq (mn^2)^k$. Since $m \in \mathcal{O}(n)$, we thus have $\rho \in \mathcal{O}(n^{3k})$.

We can now upper bound the size of T and Q . Since each entry of T stores a single bit, we immediately have that T has $\mathcal{O}(n^{3k})$ size. Instead, each entry of table Q stores $\mathcal{O}(n^k)$ keys of size $\mathcal{O}(k \log n)$; thus, Q has $\mathcal{O}(kn^{4k} \log n)$ size. The upper bound on the number of keys comes from the number of ways to map the tails of the at most k edges incoming into v_τ on the points of S , which has size n ; this number is $\binom{n}{k} \in \mathcal{O}(n^k)$. The upper bound on the size of each key comes from the fact that it consists of at most k triplets each containing an identifier of $\mathcal{O}(n)$ edges and two identifiers of $\mathcal{O}(n)$ points. $\square$

Claim A.1.6. *Tables T and Q can be computed in $\mathcal{O}(n^{4k})$ and $\mathcal{O}(kn^{4k} \log n)$ time, respectively.*

Proof. We determine the time needed to compute, for each key χ , the value $T[\chi]$ and $Q[\chi]$. For each key χ , we need to verify whether the h segments $\overline{p_i q_i}$ intersect at a point different from a common endpoint, which can be tested in $\mathcal{O}(k \log k)$ time [260]. Moreover, if $|V(G_\chi)| > 1$, computing $T[\chi]$ requires accessing the value of up to $|S_\downarrow|^{|H^+|} < n^k$ entries of T , and verifying whether at least one of them contains the value **True**. Since $n > k$, the $\mathcal{O}(k \log k)$ term in the running time is dominated by the $\mathcal{O}(n^k)$ term, hence the time needed to compute each entry $T[\chi]$ is thus $\mathcal{O}(n^k)$. Since, by Claim A.1.5, there are $\mathcal{O}(n^{3k})$ keys χ , it follows that T can be computed in overall $\mathcal{O}(n^{4k})$ time. On the other hand, the time needed to compute each entry $Q[\chi]$ is upper bounded by the time needed to write the $\mathcal{O}(n^k)$ keys contained in $Q[\chi]$, each of which has $\mathcal{O}(k \log n)$ size, i.e., $\mathcal{O}(kn^k \log n)$ time per entry. It follows that Q can be computed in overall $\mathcal{O}(kn^{4k} \log n)$ time. $\square$

Finally, recall that in order to verify whether G admits an UPSE on S , we need to check whether $\Lambda \neq \emptyset$. Computing the maximum size of an st -cutset of a planar st -graph G can be done in linear time, as it reduces to the problem of computing the length of a shortest path in the dual of any embedding of G (between the vertices representing the left and right outer faces of this embedding) [261, 248]. Therefore, the overall running time to test whether G admits an UPSE on S is dominated by the time needed to compute T , that is, $\mathcal{O}(n^{4k})$ time.

To obtain an UPSE Γ of G on S , if any, we proceed as follows. Suppose that the algorithm terminates with a positive answer and let σ be any key in Λ . We initialize Γ to a drawing of the edges in $E(\sigma)$, where each edge $e_i \in E(\sigma)$

is drawn as in σ . Then, in $\mathcal{O}(n^k)$ time, we can search in T a key χ with $T[\chi] = \text{True}$ such that σ stems from χ , and update Γ to include a drawing of the edges in $E(\chi) \setminus E(\sigma)$, where each edge $e_i \in E(\chi) \setminus E(\sigma)$ is drawn as in χ ; note that the edges in $E(\chi) \cap E(\sigma)$ are drawn in χ as they are drawn in σ . Iteratively apply such a procedure until a key α is reached such that $T[\alpha] = \text{True}$ and $E(\alpha)$ is the set of edges incident to s . This yields the desired UPSE of G on S . As the depth of the recursion is linear in the size of G , we can therefore compute Γ in $\mathcal{O}(n^{k+1})$ time.

From the above discussion, we have the following theorem.

Theorem A.1.7. *Let G be an n -vertex planar st -graph whose maximum st -cutset has size k and let S be a set of n points. UPSE TESTING can be solved for (G, S) in $\mathcal{O}(n^{4k})$ time and $\mathcal{O}(n^{3k})$ space; if an UPSE of G on S exists, it can be constructed within the same bounds.*

We now turn our attention to the design of an algorithm for the enumeration of the UPSEs of G on S . The algorithm exploits the table Q and the set Λ . By [Claims A.1.5](#) and [A.1.6](#), these can be computed in $\mathcal{O}(kn^{4k} \log n)$ time and space. Our enumeration algorithm defines and explores an acyclic digraph $\mathcal{D}$. The nodes of the digraph correspond to the keys χ of the dynamic programming table Q such that $Q[\chi] \neq \emptyset$, plus a source n_S and a sink n_T . Let χ_i and χ_j be two keys of Q such that $Q[\chi_i] \neq \emptyset$ and $Q[\chi_j] \neq \emptyset$, and let $n(\chi_i)$ and $n(\chi_j)$ be the nodes corresponding to χ_i and χ_j in $\mathcal{D}$, respectively. There exists an edge directed from $n(\chi_i)$ to $n(\chi_j)$ in $\mathcal{D}$ if $\chi_j \in Q[\chi_i]$. Also, there exists an edge directed from n_S to each node $n(\sigma)$ such that $\sigma \in \Lambda$. Finally, there exists an edge directed to n_T from each node $n(\chi)$ such that $Q[\chi] = \{\perp\}$. Note that n_S is the unique source of $\mathcal{D}$, n_T is the unique sink of $\mathcal{D}$, and $\mathcal{D}$ has no directed cycle. Hence, $\mathcal{D}$ is an $n_S n_T$ -graph.

The exploration of $\mathcal{D}$ performed by our enumeration algorithm is a depth-first traversal, in which every distinct path from n_S to n_T corresponds to an UPSE of G on S . We initialize a current UPSE Γ of G on S as $\Gamma = S$ (where no edge of G is drawn). When the visit traverses an edge of $\mathcal{D}$ directed from a node $n(\chi_i)$ to a node $n(\chi_j)$, it adds to Γ the edges in $E(\chi_j) \setminus E(\chi_i)$, drawn as in χ_j . Note that these are all the edges in $E(\chi_j)$ if $n(\chi_i) = n_S$ and it is an empty set if $n(\chi_j) = n_T$. Whenever the traversal reaches n_T , it outputs the constructed UPSE Γ of G on S . When the visit backtracks on a node $n(\chi_i)$ coming from an edge $(n(\chi_i), n(\chi_j))$, it removes from Γ the edges in $E(\chi_j) \setminus E(\chi_i)$.

To prove the correctness of the algorithm, we show what follows:

- (i) Distinct $n_S n_T$ -paths in $\mathcal{D}$ correspond to different UPSEs of G on S .

- (ii) For each UPSE of G on S , there exists in $\mathcal{D}$ an $n_S n_{\mathcal{T}}$ -path corresponding to it.

For a directed path $\mathcal{P}$ in $\mathcal{D}$, let $E(\mathcal{P})$ be the set that contains all the edges in the sets $E(\chi)$, where χ is any key corresponding to a node in $\mathcal{P}$.

- To prove **Item i**, we proceed by contradiction. Let Γ be an UPSE of G on S that is generated twice by the algorithm, when traversing distinct $n_S n_{\mathcal{T}}$ -paths $\mathcal{P}_1$ and $\mathcal{P}_2$. Let $n(\chi_x)$ be the closest node to n_S in $\mathcal{P}_1$ and $\mathcal{P}_2$ such that $(n(\chi_x), n(\chi_1))$ is an edge in $\mathcal{P}_1$ and $(n(\chi_x), n(\chi_2))$ is an edge in $\mathcal{P}_2$, with $n(\chi_1) \neq n(\chi_2)$, where χ_x , χ_1 , and χ_2 are keys of Q . Note that, since the path $\mathcal{P}_x$ from n_S to $n(\chi_x)$ (possibly such a path is a single node if $n_S = n(\chi_x)$) is the same in $\mathcal{P}_1$ and $\mathcal{P}_2$, the restriction Γ_x of Γ to the edge set $E(\mathcal{P}_x)$ is also the same in $\mathcal{P}_1$ and $\mathcal{P}_2$. Hence, the tail v_{χ_x} with largest y -coordinate of an edge in $E(\chi_x)$ is uniquely defined by Γ_x . This implies that the edge sets $E(\chi_1)$ and $E(\chi_2)$ coincide, as they are both obtained from $E(\chi_x)$ by replacing the edges outgoing from v_{χ_x} with the edges incoming into v_{χ_x} in G . Since $E(\chi_1) = E(\chi_2)$ and $\chi_1 \neq \chi_2$, it follows that χ_1 and χ_2 must differ in the way such keys map the tails of the edges incoming into v_{χ_x} to the points of S . Then the UPSEs yielded by $\mathcal{P}_1$ and $\mathcal{P}_2$ are different, a contradiction.
- To prove **Item ii**, we show that, if there exists an UPSE Γ of G on S , then there exists a path in $\mathcal{D}$ from n_S to $n_{\mathcal{T}}$ that yields Γ . For $i = 1, \dots, n$, let S_i be the set that consists of the lowest i points of S . Also, for $i = 1, \dots, n-1$, let Γ_i be the restriction of Γ to the vertices of G mapped to S_i and to all their incident edges, including those whose other end-vertex is not in S_i . We claim that there exists a path $\mathcal{P}_i$ in $\mathcal{D}$ that starts from a node n_i and ends at $n_{\mathcal{T}}$ such that: (1) the set $E(\mathcal{P}_i)$ coincides with the set of edges that are embedded in Γ_i ; (2) the embedding of the edges in $E(\mathcal{P}_i)$ defined by the keys χ corresponding to nodes in $\mathcal{P}_i$ is the same as in Γ_i ; and (3) let χ_i be the key corresponding to n_i , then $E(\chi_i)$ contains all and only the edges e of G such that an end-vertex of e is mapped by Γ to a point in S_i and the other end-vertex of e is mapped by Γ to a point not in S_i . The claim implies **Item ii**, as when $i = n-1$, we have that $E(\mathcal{P}_{n-1})$ is the edge set of G , by (1), and that the embedding of the edges in $E(\mathcal{P}_{n-1})$ defined by the keys χ corresponding to nodes in $\mathcal{P}_{n-1}$ is Γ , by (2), hence $(n_S, \chi_{n-1}) \cup \mathcal{P}_{n-1}$ is the desired path from n_S to $n_{\mathcal{T}}$ that yields Γ .

In order to prove the claim, we proceed by induction. In the base case, we have $i = 1$, hence S_1 consists only of the point p_s and Γ_1 is the restriction of Γ to all the edges incident to s . Since Γ is an UPSE, Γ_1 is an embedding of such edges in which s lies on p_s and any two heads of such edges are not aligned with p_s . Hence, by construction, there is a key χ such that $E(\chi)$ consists of the set of edges incident to s , such that $Q[\chi] = \{\perp\}$, and such that the embedding of the edges in $E(\chi)$ on S defined by χ is Γ_1 . It follows that $\mathcal{D}$ contains a node $n(\chi)$ corresponding to χ , and thus a path $\mathcal{P}_1 = (n(\chi), n_{\mathcal{T}})$ with the properties required by the claim.

For the inductive case, we have $i > 1$. Let p_i be the point of S_i with highest y -coordinate and let v_i be the vertex of G mapped to p_i by Γ . By induction, there exists a path $\mathcal{P}_{i-1}$ in $\mathcal{D}$ that starts from a node n_{i-1} and ends at $n_{\mathcal{T}}$ such that: (1) the set $E(\mathcal{P}_{i-1})$ is the set of edges embedded in Γ_{i-1} ; (2) the embedding of the edges in $E(\mathcal{P}_{i-1})$ defined by the keys corresponding to nodes in $\mathcal{P}_{i-1}$ defines Γ_{i-1} ; and (3) let χ_{i-1} be the key corresponding to n_{i-1} , then $E(\chi_{i-1})$ contains all and only the edges whose end-vertices are mapped by Γ one to a point in S_{i-1} and the other to a point not in S_{i-1} . Note that (3) ensures that all the edges incoming into v_i are in $E(\chi_{i-1})$.

Consider the edge set H_i composed of the edges outgoing from v_i and of the edges in $E(\chi_{i-1})$, except for those incoming into v_i . We prove that H_i is an *st*-cutset. Indeed, by (3), every edge of G that in Γ starts from a point below p_i and ends at a point above p_i is in $E(\chi_{i-1})$. Then H_i comprises all the edges that start from p_i or from a point below p_i and end at a point above p_i . Hence, the removal of the edges of H_i splits G into two connected subgraphs, one induced by the vertices (including s) mapped by Γ to S_i , and one induced by the vertices (including t) mapped by Γ to the points above p_i .

Since H_i is an *st*-cutset, there exists a key χ_i such that $E(\chi_i) = H_i$ and the edges of $E(\chi_i)$ are embedded in χ_i as in Γ_i . Note that p_i is the tail of an edge in $E(\chi_i)$ with largest y -coordinate, hence our algorithm, starting from the *st*-cutset $E(\chi_i)$, removes the edges outgoing from v_i , and adds the edges incoming into v_i , thus it constructs the *st*-cutset $E(\chi_{i-1})$ and, from there, the key χ_{i-1} in which the edges of $E(\chi_{i-1})$ are mapped as in Γ_{i-1} . The algorithm then inserts χ_{i-1} in $Q[\chi_i]$, and hence the digraph $\mathcal{D}$ contains the edge (n_i, n_{i-1}) , where n_i is the node of $\mathcal{D}$ corresponding to χ_i . This completes the induction, hence the proof of the claim and the one of [Item ii](#).

It remains to discuss the running time of our enumeration algorithm. Concerning the set-up time, the table Q can be constructed in $\mathcal{O}(kn^{4k} \log n)$ time, by [Claim A.1.6](#). Also, the digraph $\mathcal{D}$ can be constructed in linear time in the size of Q , which is $\mathcal{O}(kn^{4k} \log n)$ by [Claim A.1.5](#); indeed, the edges outgoing from a node $n(\chi)$ in $\mathcal{D}$ are those toward the nodes whose corresponding keys are in $Q[\chi]$. Concerning the space usage, again by [Claim A.1.5](#), we have that Q and $\mathcal{D}$ have $\mathcal{O}(kn^{4k} \log n)$ size. Finally, we discuss the delay of our algorithm. The paths from n_S to n_T have $\mathcal{O}(n)$ size. Between an UPSE and the next one, at most two paths are traversed (one to backtrack and one to again reach n_T). The total number of edges which are deleted from or added to the current embedding when traversing such paths is in $\mathcal{O}(n)$, hence the delay of our algorithm is $\mathcal{O}(n)$. We get the following.

Theorem A.1.8. *Let G be a n -vertex planar st -graph whose maximum st -cut has size k and let S be a set of n points. It is possible to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time.*

A.1.5 Planar st -Graphs Composed of Two st -Paths

In this section, we discuss a special, and in our opinion interesting, case of [Theorem A.1.7](#), namely the one in which the underlying graph of the given planar st -graph is an n -vertex cycle. Applying [Theorem A.1.7](#) to this setting would yield an $\mathcal{O}(n^8)$ -time testing algorithm. Now, based on a characterization of the positive instances, we give a much faster algorithm for this case, provided that the points of S are in general position.

Theorem A.1.9. *Let G be an n -vertex planar st -graph consisting of two st -paths P_L and P_R , and let S be a pointset with n points in general position. We have that G admits an UPSE on S with P_L to the left of P_R if and only if $|P_L| \geq |\mathcal{H}_L(S)|$ and $|P_R| \geq |\mathcal{H}_R(S)|$. Also, it can be tested in $\mathcal{O}(n \log n)$ time whether G admits an UPSE on S .*

Proof. Provided the characterization in the statement holds, we can easily test whether G admits an UPSE on S by computing the convex hull $\mathcal{CH}(S)$ of S (which can be done in $\mathcal{O}(n \log n)$ time), deriving the sets $\mathcal{H}_L(S)$ and $\mathcal{H}_R(S)$ (which can be done in $\mathcal{O}(n)$ time, by scanning $\mathcal{CH}(S)$), and finally comparing their sizes with the ones of P_L and P_R (which can be done in $\mathcal{O}(1)$ time). Therefore, in the following we focus on proving the characterization.

For the necessity, suppose for a contradiction that there exists an UPSE on S with P_L to the left of P_R and that $|P_L| < |\mathcal{H}_L(S)|$; the case in which $|P_R| < |\mathcal{H}_R(S)|$ is analogous. Since $|P_L \cup P_R| = n$, a vertex d of P_R must be drawn on a point in $\mathcal{H}_L(S)$. Consider the subpath P_d of P_R between s and d . The drawing of P_d splits $\mathcal{CH}(S)$ into two closed regions, to the left and to the right of P_d . In any UPSE of G on S with P_L to the left of P_R , we have that P_L lies in both regions, namely it lies in the region to the left of P_d with the edge incident to s and it lies in the region to the right of P_d at t . Hence, the drawing of P_L crosses the drawing of P_d , and thus the one of P_R , a contradiction.

In the following, we prove the sufficiency by induction on the size of S (and, thus, of $V(G)$). We give some preliminary definitions; see [Figures A.1.5 to A.1.7](#). Let p_s and p_t be the south and north extreme of S , respectively. Consider the line ℓ_{st} through p_s and p_t . Let S_L (S_R) be the set consisting of the points of S lying in the closed half-plane delimited by ℓ_{st} that includes all points that lie to the left (resp. right) of ℓ_{st} , including p_s and p_t . Note that $\mathcal{H}_L(S) \subseteq S_L$ and $\mathcal{H}_R(S) \subseteq S_R$. Moreover, since S is in general position, it holds that $S_L \cap S_R = \{p_s, p_t\}$.

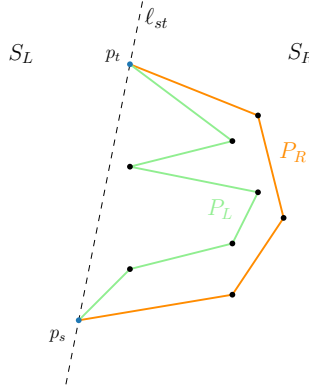


Figure A.1.5: Illustration for the base case of [Theorem A.1.9](#).

In the base case, it either holds that (a) $S_L = \{p_s, p_t\}$ and $|\mathcal{H}_R(S)| = |P_R|$, or (b) $S_R = \{p_s, p_t\}$ and $|\mathcal{H}_L(S)| = |P_L|$. We discuss the former case (see [Figure A.1.5](#)), as the latter case is symmetric. In this case, an UPSE Γ of G on S clearly exists and is, in fact, unique. In particular, the drawing of P_R in Γ coincides with the right envelope $\mathcal{E}_R(S)$ of S , while the drawing of P_L in Γ is the y -monotone polyline that assigns to the j -th internal vertex of P_L

(when traversing P_L from s to t) the point of $S_R \setminus \mathcal{H}_R(S)$ with the j -th smallest y -coordinate. Since each of such paths is y -monotone, it is not self-crossing. Also, no edge of P_L crosses an edge of P_R , as the drawing of P_R in Γ coincides with $\mathcal{E}_R(S)$.

If the base case does not hold, then we distinguish two cases based on whether both S_L and S_R contain a vertex different from p_s and p_t (**Case A**), or only one of them does (**Case B**). In the following, we assume that in **Case B** the set S_R contains a vertex different from p_s and p_t , the case in which only S_L contains a vertex different from p_s and p_t can be treated symmetrically. More formally, in **Case A** we have that $\{p_s, p_t\} \subset S_L$ and $\{p_s, p_t\} \subset S_R$, whereas in **Case B** we have that $S_L = \{p_s, p_t\}$ and $\{p_s, p_t\} \subset S_R$. Note that, in **Case B**, since the conditions of the base case do not apply and by the hypothesis of the statement, we have that $|P_R| > |\mathcal{H}_R(S)|$ holds.

If **Case A** holds, we distinguish two subcases. In **Case A1**, it holds $|P_L| \geq |S_L|$, whereas in **Case A2**, it holds $|P_L| < |S_L|$. We discuss **Case A1** (see [Figure A.1.6](#)); **Case A2** can be treated symmetrically, given that in this case it holds that $|P_R| \geq |S_R|$.

Suppose that **Case A1** holds true. Then $\mathcal{H}_L(S)$ contains a point p different from p_s and p_t ; see [Figure A.1.6a](#). Since by the hypotheses of this case $|P_L| \geq |S_L| \geq |\mathcal{H}_L(S)|$ and $|\mathcal{H}_L(S)| \geq 3$, we have that P_L contains at least one internal vertex. Let $S' = S \setminus \{p\}$, let P'_L be an st -path with $|P'_L| = |P_L| - 1$, and let G' be the planar st -graph $P'_L \cup P_R$. Since $|\mathcal{H}_L(S')| \leq |S_L| - 1$ and since $|S_L| \leq |P_L|$, we have that $|\mathcal{H}_L(S')| \leq |P_L| - 1 = |P'_L|$. Thus, the graph G' and the pointset S' satisfy the conditions of the statement. By induction, we have

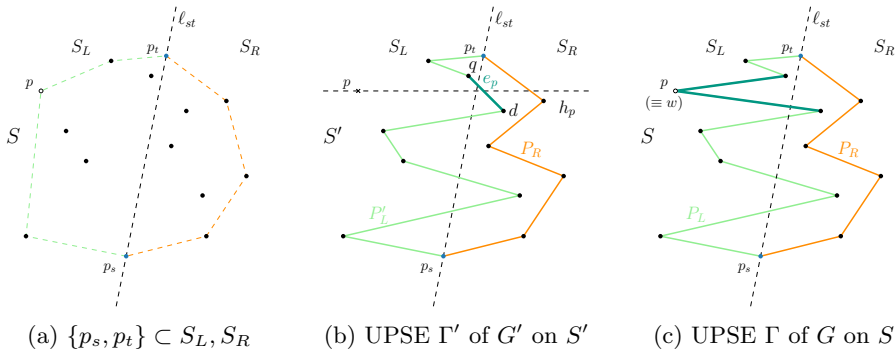
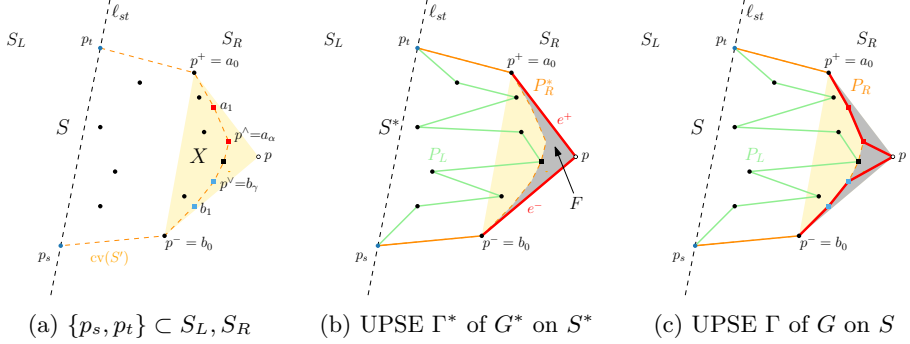


Figure A.1.6: Illustrations for **Case A** in the proof of [Theorem A.1.9](#).


 Figure A.1.7: Illustrations for **Case B2** in the proof of **Theorem A.1.9**.

that G' admits an UPSE Γ' on S' , see **Figure A.1.6b**.

We show how to modify Γ' to obtain an UPSE Γ of G on S as follows; see **Figures A.1.6b** and **A.1.6c**. The drawing of P_R is the same in Γ as in Γ' . Let h_p be the horizontal line passing through p . Since Γ' is an UPSE of G' on S' and since $y(p_s) < y(p) < y(p_t)$, we have that h_p intersects the drawing of P'_L in a single point. Such a point belongs to a segment that is the image of an edge e_p of P'_L . Let d and q be the extremes of such a segment that are the images of the tail and of the head of e_p in Γ' , respectively. We show how to modify the drawing of P'_L to obtain a y -monotone drawing of P_L that does not intersect P_R . To this aim, we replace the drawing of e_p with the y -monotone polyline composed of the segments $\overline{dp}$ and $\overline{pq}$. Note that such a polyline lies in the interior of the region delimited by the segment $\overline{dq}$ (representing e_p) and by the horizontal rays originating at d and q and directed leftward. Due to the fact that P'_L is represented as a y -monotone polyline in Γ' , such a region is not traversed by the drawing of any edge. Thus, Γ is an UPSE of G on S . We refer to the described procedure as the *p -leftward-outer-extension* of Γ' ; a *p -rightward-outer-extension* of Γ' is defined symmetrically.

If **Case B** holds, recall that $S_L = \{p_s, p_t\} \subset S_R$, and since the base case does not apply, we have that $|P_R| > |\mathcal{H}_R(S)|$. Let p be any point in $\mathcal{H}_R(S) \setminus \{p_s, p_t\}$ and $S' = S \setminus \{p\}$. By the conditions of **Case B**, the path P_R contains at least one internal vertex. We let P'_R be an st -path with $|P'_R| = |P_R| - 1$, and we let G' be the st -graph $P_L \cup P'_R$. We distinguish two cases based on the size of $\mathcal{H}_R(S')$. In **Case B1**, it holds $|P'_R| \geq |\mathcal{H}_R(S')|$, whereas in **Case B2**, it holds $|P'_R| < |\mathcal{H}_R(S')|$.

In **Case B1**, the pair (G', S') satisfies the conditions of the statement. In

particular, it either matches the conditions of the base case or again those of **Case B**. Thus, since $|S'| = |S| - 1$ (and $|V(G')| = |V(G)| - 1$), we can inductively construct an UPSE Γ' of G' on S' , and obtain an UPSE of G on S via a p -rightward-outer-extension of Γ' .

In **Case B2**, which is the most interesting, we proceed as follows; see [Figure A.1.7](#). Let p^+ be the point of $\mathcal{H}_R(S)$ with the smallest y -coordinate and above p and let p^- be the point of $\mathcal{H}_R(S)$ with the largest y -coordinate and below p . Let X be the set of points of S that lie in the interior of the triangle Δp^+pp^- , including p^+ and p^- and excluding p . Clearly, the right envelope of $\mathcal{CH}(X)$ forms a subpath of the right envelope of $\mathcal{CH}(S')$. The set $\mathcal{H}_R(X)$ consists of p^- , p^+ , and of k vertices not belonging to $\mathcal{H}_R(S)$, depicted as squares in [Figure A.1.7a](#). Denote by $k^* = |P_R| - |\mathcal{H}_R(S)|$ the number of points in the interior of $\mathcal{CH}(S)$ that need to be the image of a vertex of P_R in an UPSE of G on S . Observe that $k > k^* > 0$ holds true. Indeed, $k^* > 0$ holds true since (G, S) does not satisfy the conditions of the base case, and $k > k^*$ holds true since (G, S) does not satisfy the conditions of **Case B1**. Let $p^\wedge$ be the point of $\mathcal{H}_R(S')$ with the smallest y -coordinate and above p , and let $p^\vee$ be the point with the largest y -coordinate and below p . Up to renaming, let $a_0 = p^+$, $a_1, \dots, a_\alpha = p^\wedge$ be the subsequence of points of $\mathcal{E}_R(X)$ encountered when traversing $\mathcal{E}_R(X)$ from p^+ to $p^\wedge$ and observe that these points have decreasing y -coordinates. Similarly, let $b_0 = p^-$, $b_1, \dots, b_\gamma = p^\vee$ be the subsequence of points of $\mathcal{E}_R(X)$ encountered when traversing $\mathcal{E}_R(X)$ from p^- to $p^\vee$ and observe that these points have increasing y -coordinates. We define the set $X^* \subset \mathcal{H}_R(X)$ as follows. If $k^* \leq \alpha$, then let $X_\wedge^* = \{a_i | 1 \leq i \leq k^*\}$ and $X_\vee^* = \emptyset$, otherwise let $X_\wedge^* = \{a_i | 1 \leq i \leq \alpha\}$ and $X_\vee^* = \{b_i | 1 \leq i \leq k^* - \alpha\}$. In both cases, let $X^* = X_\wedge^* \cup X_\vee^*$. Observe that $|X^*| = k^*$. Also, by the definition of k^* , the path P_R contains $|\mathcal{H}_R(S)| - 2 + k^*$ internal vertices and since $|\mathcal{H}_R(S)| \geq 3$ in **Case B**, we have that P_R contains at least $k^* + 1$ internal vertices.

Let $S^* = S \setminus X^*$, let P_R^* be an st -path with $|P_R| - k^*$ vertices, and let G^* be the st -graph $P_L \cup P_R^*$. We have that the pair (G^*, S^*) satisfies the conditions of the statement, and in particular the base case. In fact, $|P_R^*| = |P_R| - k^*$, and by the definition of k^* , we have that $|P_R| - k^* = |\mathcal{H}_R(S)|$. Moreover, by construction, $\mathcal{H}_R(S) = \mathcal{H}_R(S^*)$, since the vertices of X^* lie in the interior of $\mathcal{CH}(S)$. Thus, since $|S^*| = |S| - k^*$, by induction G^* admits an UPSE Γ^* on S^* ; see [Figure A.1.7b](#).

We now show how to transform Γ^* into an UPSE Γ of G on S . Since the base case applies to (G^*, S^*) , we have that the endpoints of the edges of P_R^* are consecutive along $\mathcal{E}_R(S)$. In particular, there exist two adjacent edges e^-

and e^+ of P_R^* such that the tail of e^- is mapped to p^- , the head of e^- , which is the tail of e^+ , is mapped to p , and the head of e^+ is mapped to p^+ . Therefore, the UPSE Γ of G on S can be obtained from Γ^* as follows; see [Figure A.1.7c](#). We initialize $\Gamma = \Gamma^*$. The drawing of P_L is the same in Γ as in Γ^* . Next, we show how to modify the drawing of P_R^* to obtain a y -monotone drawing P_R that does intersect the drawing of P_L and uses the same points as P_L^* and the points in X^* . To this aim, we replace the drawing of e^+ with the (unique) y -monotone polyline connecting p and p^+ that passes through all the points in $X_\wedge^*$. Also, we replace the drawing of e^- with the (unique) y -monotone polyline connecting p^- and p that passes through all the points in $X_\vee^*$; note that $X_\vee^*$ might be empty, in which case the polyline still coincides with the drawing of e^- . This concludes the construction of Γ . To see that Γ is an UPSE of G on S observe that the above polylines (i) are each non-self-crossing, as they are y -monotone, (ii) do not cross with each other as they entirely lie either above or below p (and only meet at p), and (iii) do not cross any edge of Γ' as they lie in the region F (shaded gray in [Figures A.1.7b](#) and [A.1.7c](#)) obtained by subtracting from the triangle Δp^+pp^- (interpreted as a closed region) all the points of $\mathcal{CH}(X)$. Indeed, observe that in Γ^* , the region F is not traversed by any edge and that the only points of S^* that lie on the boundary of F are p and the points in $\mathcal{H}_R(X) \setminus X^*$. $\square$

A.1.6 Enumerating Non-crossing Monotone Hamiltonian Cycles

[Theorem A.1.9](#) allows us to test whether an n -vertex planar *st*-graph G composed of two *st*-paths can be embedded as a non-crossing monotone Hamiltonian cycle on a set S of n points. We now show an efficient algorithm for enumerating *all* the non-crossing monotone Hamiltonian cycles on S . [Figure A.1.8](#) shows two non-crossing monotone Hamiltonian cycles on a pointset.

Theorem A.1.10. *Let S be a set of n points. It is possible to enumerate all the non-crossing monotone Hamiltonian cycles on S with $\mathcal{O}(n)$ delay, using $\mathcal{O}(n^2)$ space, after $\mathcal{O}(n^2)$ set-up time.*

Let $p_1, \dots, p_n$ be the points of S , ordered by increasing y -coordinates. This order can be computed in $\mathcal{O}(n \log n)$ time. For $i \in [n]$, let $S_i = \{p_1, \dots, p_i\}$. A *bipath* B on S_i consists of two non-crossing monotone paths L and R on S_i , each of which might be a single point, such that (see [Figure A.1.9](#)):

- (i) L and R start at p_1 ;

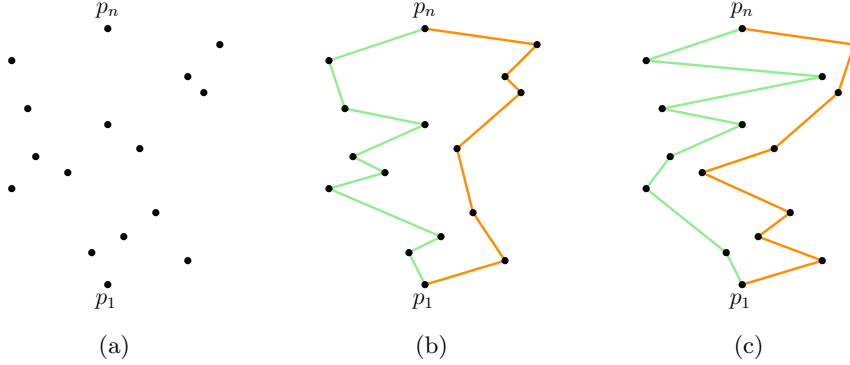


Figure A.1.8: Two non-crossing monotone Hamiltonian cycles on the same pointset.

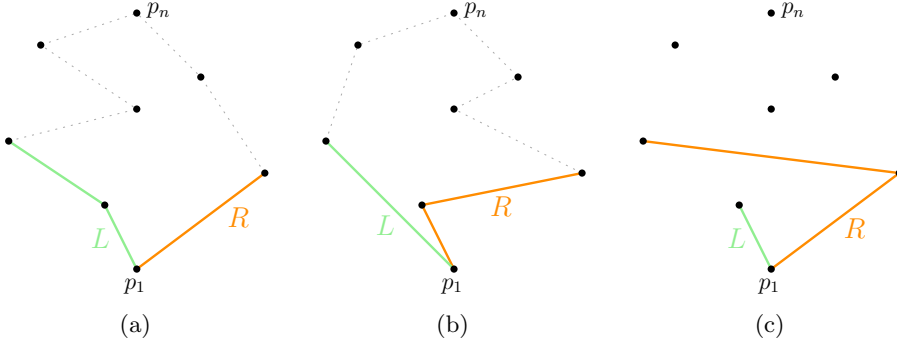


Figure A.1.9: Three bipaths on S_4 . The first two bipaths are extendible, while the third one is not. Dotted lines complete non-crossing Hamiltonian cycles on S whose restriction to S_4 is the bipath.

- (ii) each point of S_i is the image of an endpoint of a segment of B ; and
- (iii) if L and R both have at least one segment, then L is to the left of R .

We say that a bipath B is *extendible* if there exists a non-crossing monotone Hamiltonian cycle on S whose restriction to S_i is B . Consider a bipath B on S_i with $1 < i < n$. Let $p_{\ell(B)}$ and $p_{r(B)}$ be the endpoints of L and R with the highest y -coordinate, respectively. Suppose first that $\ell(B) > r(B)$, that is, $p_{\ell(B)}$ is higher than $p_{r(B)}$. Then note that $\ell(B) = i$; also, it might be

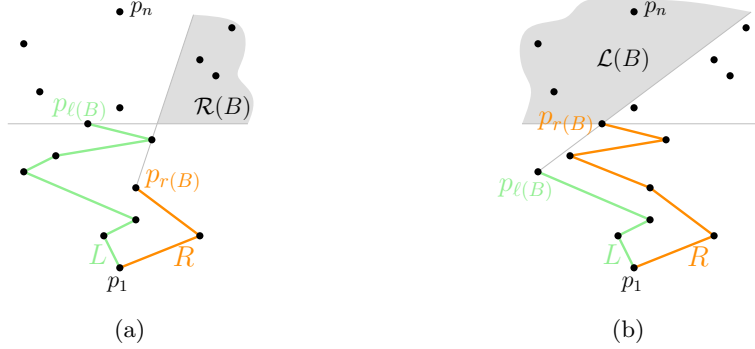


Figure A.1.10: (a) Region $\mathcal{R}(B)$ for a bipath B with $\ell(B) > r(B)$. (b) Region $\mathcal{L}(B)$ for a bipath B with $r(B) > \ell(B)$.

that $r(B) = 1$. Consider the ray $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$; recall that this is the rightmost ray starting at $p_{r(B)}$ and passing through a point of $S_{\ell(B)} \setminus S_{r(B)}$. We denote by $\mathcal{R}(B)$ the open region of the plane strictly to the right of $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$ and strictly above the horizontal line through $p_{\ell(B)}$; see [Figure A.1.10a](#). Similarly, if $p_{r(B)}$ is higher than $p_{\ell(B)}$, then $\mathcal{L}(B)$ is the open region of the plane strictly to the left of the leftmost ray $\ell(p_{\ell(B)}, S_{r(B)} \setminus S_{\ell(B)})$ from $p_{\ell(B)}$ through a point of $S_{r(B)} \setminus S_{\ell(B)}$ and strictly above the horizontal line through $p_{r(B)}$; see [Figure A.1.10b](#).

For any $i \in [n - 1]$, we say that a bipath B on S_i is *safe* if:

- (i) $i = 1$; or
- (ii) $i > 1$, $p_{\ell(B)}$ is higher than $p_{r(B)}$, and $|\mathcal{R}(B) \cap S| \geq 1$; or
- (iii) $i > 1$, $p_{r(B)}$ is higher than $p_{\ell(B)}$, and $|\mathcal{L}(B) \cap S| \geq 1$.

We have the following lemma.

Lemma A.1.11. *A bipath B is extensible if and only if it is safe.*

Proof. First, we prove the necessity. Suppose that B is extensible and let C be any non-crossing monotone Hamiltonian cycle on S whose restriction to S_i is B . Also suppose, for a contradiction, that B is not safe, which implies that $i > 1$. Assume that $p_{\ell(B)}$ is higher than $p_{r(B)}$, as the other case is symmetric. Then we have $\mathcal{R}(B) \cap S = \emptyset$. Let $\overline{p_{r(B)}p'_{r(B)}}$ be the segment of C such that



Figure A.1.11: Since the point p on the ray $\rho(p_{r(B)}, S \setminus S_i)$ defines a segment $\overline{p_{r(B)}p}$ which is on the boundary of the convex hull of $S \setminus S_{r(B)-1}$ (the convex hull is shaded light-gray), we can complete R via the boundary of the convex hull and L via the remaining points.

$y(p'_{r(B)}) > y(p_{r(B)})$. Since all points in $S_{\ell(B)} \setminus S_{r(B)}$ belong to L , we have $p'_{r(B)}$ lies strictly above the horizontal line through $p_{\ell(B)}$. This, together with the fact that S contains no point strictly above the horizontal line through $p_{\ell(B)}$ and to the right of the ray $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$, implies that the ray $\rho(p_{r(B)}p'_{r(B)})$ lies to the left of the ray $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$, which implies that the segment $(p_{r(B)}, p'_{r(B)})$ crosses the path L , a contradiction to the fact that C is non-crossing.

Second, we prove the sufficiency. Suppose that B is safe. We show how to construct a non-crossing monotone Hamiltonian cycle C on S whose restriction to S_i is B . Assume first that $i = 1$. Then C can be constructed as the union of two monotone paths. The first path is one of the two paths between s and t on the boundary of the convex hull of S . The second path from s to t traverses all the points of S that are not on the first path, in increasing order of y -coordinate. Assume next that $i > 1$ and refer to [Figure A.1.11](#). Assume also that $p_{\ell(B)}$ is higher than $p_{r(B)}$, as the other case is symmetric. Then $\mathcal{R}(B)$ contains some points of S . Consider the rightmost ray $\rho(p_{r(B)}, S \setminus S_i)$ starting from $p_{r(B)}$ and passing through a point p in $S \setminus S_i$. Observe that $\overline{p_{r(B)}p}$ is a segment on the boundary of the convex hull of $S \setminus S_{r(B)-1}$. Hence, we can augment R so that it becomes a monotone path from s to t , by adding to it the part of the boundary of the convex hull of $S \setminus S_{r(B)-1}$ from $p_{r(B)}$ to t (by proceeding in counter-clockwise direction on this boundary from $p_{r(B)}$ to t). Also, we can augment L so that it becomes a monotone path from s to t by making it pass through all the points in $S \setminus S_i$ that are not used by R , and finishing at t . $\square$

We now describe our algorithm. The algorithm implicitly defines and explores a search tree T . The leaves of T have level n and correspond to non-crossing monotone Hamiltonian cycles on S . The internal nodes at level i correspond to extensible bipaths on S_i and have at most two children each. The exploration of T performed by our enumeration algorithm is a depth-first traversal. When a node μ is visited, the number of its children is established. If μ has at least one child, the visit proceeds with any child of μ . Otherwise, μ is a leaf; then the visit proceeds with any unvisited child of the ancestor of μ that has largest level, among the ancestors of μ that have unvisited children.

- The algorithm starts at the root of T , which corresponds to the (unique) safe bipath on S_1 .
- At each node μ at level $i \in [n - 2]$ of T , corresponding to a bipath $B(\mu)$, we construct either one or two bipaths on S_{i+1} , associated with either one or two children of μ , respectively. Let $L(\mu)$ and $R(\mu)$ be the left and right non-crossing monotone paths composing $B(\mu)$, respectively, and let $p_{\ell(B(\mu))}$ and $p_{r(B(\mu))}$ be the endpoints of $L(\mu)$ and $R(\mu)$ with the highest y -coordinate, respectively. If $\overline{p_{\ell(B(\mu))}p_{i+1}}$ does not cross $R(\mu)$, then let $B_L = B(\mu) \cup \overline{p_{\ell(B(\mu))}p_{i+1}}$. We test whether B_L is a safe bipath and, in the positive case, add to μ a child μ_L corresponding to B_L . Analogously, if $\overline{p_{r(B(\mu))}p_{i+1}}$ does not cross $L(\mu)$, then we test whether $B_R = B(\mu) \cup \overline{p_{r(B(\mu))}p_{i+1}}$ is a safe bipath and, in the positive case, add to μ a child μ_R corresponding to B_R . Note that the algorithm guarantees that each node at a level smaller than or equal to $n - 1$ of T is safe, and thus, by [Lemma A.1.11](#), extensible.
- Finally, at each node μ at level $n - 1$, we add a leaf λ to μ corresponding to the non-crossing monotone Hamiltonian cycle $B(\mu) \cup \overline{p_{\ell(B(\mu))}p_n} \cup \overline{p_{r(B(\mu))}p_n}$. Note that, since μ is extensible, such a cycle is indeed non-crossing.

In order to complete the proof of [Theorem A.1.10](#), we show what follows:

- (i) Each node of T at level $i \neq n$ is internal.
- (ii) Each leaf corresponds to a non-crossing monotone Hamiltonian cycle on S .
- (iii) Distinct leaves correspond to different non-crossing monotone Hamiltonian cycles on S .

- (iv) For each non-crossing monotone Hamiltonian cycle on S , there exists a leaf of T corresponding to it.
 - (v) Using $\mathcal{O}(n^2)$ pre-processing time and $\mathcal{O}(n^2)$ space, the algorithm enumerates each non-crossing monotone Hamiltonian cycle on S with $\mathcal{O}(n)$ delay.
- To prove **Item i**, we show that the leaves of T have all level n . Consider a node μ of T with level $i < n - 1$, we prove that it has a child in T . Recall that $B(\mu)$ is safe, otherwise it would not had been added to T , and thus, by **Lemma A.1.11**, it is extensible. Hence, there exists a non-crossing monotone Hamiltonian cycle C on S whose restriction to S_i is $B(\mu)$. Also, the restriction of C to S_{i+1} is a bipath $B'(\mu)$ on S_{i+1} which coincides with $B(\mu)$, except that it contains either the segment $\overline{p_{\ell(B(\mu))}p_{i+1}}$ or the segment $\overline{p_{r(B(\mu))}p_{i+1}}$. Since $B'(\mu)$ is the restriction of C to S_{i+1} , it is extensible and thus, by **Lemma A.1.11**, it is safe. It follows that μ has a child corresponding to $B'(\mu)$, which is inserted in T when adding either the segment $\overline{p_{\ell(B(\mu))}p_{i+1}}$ or the segment $\overline{p_{r(B(\mu))}p_{i+1}}$ to $B(\mu)$. The proof that a node with level $n - 1$ is not a leaf is analogous.
 - To prove **Item ii**, consider a leaf λ and its parent μ in T . Note that μ is associated with a safe bipath $B(\mu)$ on S_{n-1} ; by **Lemma A.1.11**, we have that $B(\mu)$ is extensible. Since $B(\mu)$ is extensible, the (unique) monotone Hamiltonian cycle on S whose restriction to S_{n-1} is $B(\mu)$ is non-crossing. This cycle corresponds to λ and is added to T when visiting μ .
 - To prove **Item iii**, suppose for a contradiction that there exist two leaves λ_1 and λ_2 associated with two monotone Hamiltonian cycles C_1 and C_2 , respectively, with $C_1 = C_2$. Let μ be the lowest common ancestor of λ_1 and λ_2 in T . Let j be the level of μ . Denote by μ_i the child of μ leading to λ_i , with $i \in \{1, 2\}$. By the construction of T , we have that exactly one of the bipaths $B(\mu_1)$ and $B(\mu_2)$ contains the segment $\overline{p_{\ell(B(\mu))}p_{j+1}}$, while the other one contains the segment $\overline{p_{r(B(\mu))}p_{j+1}}$. This contradicts the fact that $C_1 = C_2$.
 - To prove **Item iv**, let C be a non-crossing monotone Hamiltonian cycle on S . Consider the safe bipath B on S_{n-1} obtained by removing from C the point p_n , together with its two incident segments. It suffices to show that T contains a node μ such that $B = B(\mu)$. In fact, in this case, μ is an extensible node of level $n - 1$ whose unique child in T is the leaf

corresponding to C . To prove that T contains such a node μ , we prove by induction that, for every level $i = 1, \dots, n - 1$, the tree T contains a node corresponding to the restriction B_i of B to S_i . The base case trivially holds. For the inductive case, suppose that T contains a node ν whose associated bipath $B(\nu)$ is B_{i-1} . Then B_i is obtained by adding either the segment $\overline{p_{\ell(B(\nu))}p_i}$ or the segment $\overline{p_{r(B(\nu))}p_i}$ to B_{i-1} . Since B_i is extensible, by [Lemma A.1.11](#) it is safe, and hence ν has a child in T corresponding to B_i .

- Finally, we prove [Item v](#). To this aim, we compute in $\mathcal{O}(n^2)$ time two tables C and D . The first one allows us to quickly test whether a bipath on S_i can be extended to a bipath on S_{i+1} (so that no crossing is introduced). The second table allows us to quickly test whether a bipath on S_i is safe.

We first describe the computation of the table C , which has $\mathcal{O}(n^2)$ size, can be computed in $\mathcal{O}(n^2)$ time, and allows us to answer in $\mathcal{O}(1)$ time the following questions: Given a bipath B on S_i composed of the monotone *st*-paths L and R respectively ending at points p_ℓ and p_r , is $B \cup \overline{p_\ell p_{i+1}}$ a bipath on S_{i+1} and is $B \cup \overline{p_r p_{i+1}}$ a bipath on S_{i+1} ? That is, the table allows us to test whether the segment $\overline{p_\ell p_{i+1}}$ crosses any edge of R and whether the segment $\overline{p_r p_{i+1}}$ crosses any edge of L .

We only discuss how C allows us to decide whether the segment $\overline{p_\ell p_{i+1}}$ crosses any edge of R , as the arguments for deciding whether the segment $\overline{p_r p_{i+1}}$ crosses any edge of L are analogous. If $i = \ell$, then obviously the segment $\overline{p_\ell p_{i+1}}$ does not cross any edge of R , as it lies completely above R . So in the following we assume that $i = r$, that is, the point p_ℓ is lower than p_r , which is the highest point of S_i . This implies that R contains the polyline $(p_{\ell+1}, p_{\ell+2}, \dots, p_r)$, as in [Figure A.1.12a](#).

A key point for our efficient test is that whether $B \cup \overline{p_\ell p_{r+1}}$ is a bipath only depends on the points $p_\ell, p_{\ell+1}, \dots, p_r, p_{r+1}$, and not on the points lower than p_ℓ . In particular, let p_x be the point of S_i with $x < \ell$ such that the segment $\overline{p_x p_{\ell+1}}$ belongs to R . Then the actual placement of p_x does not matter for whether $\overline{p_\ell p_{r+1}}$ crosses $\overline{p_x p_{\ell+1}}$ or not, see [Figure A.1.12b](#). This is formalized in the following claim.

Claim A.1.12. *Let B be a bipath on S_i composed of two monotone *st*-paths L and R ending at points p_ℓ and p_r , where $\ell < r$, and let p_x be the point of S_i with $x < \ell$ such that the segment $\overline{p_x p_{\ell+1}}$ belongs to R . Also, let q_ℓ be any point on the horizontal line h_ℓ through p_ℓ , to the right*

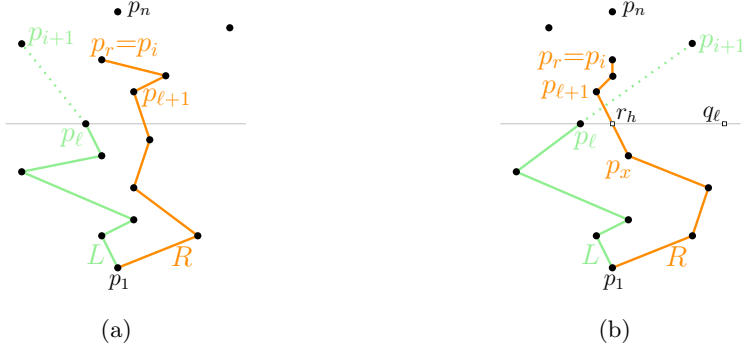


Figure A.1.12: Extensibility of a bipath B whose monotone st -paths L and R end at points p_ℓ and p_r with a segment $\overline{p_\ell p_{i+1}}$. In (a) the segment $\overline{p_\ell p_{i+1}}$ does not cross B , while in (b) it does.

of every point in S . Then the segment $\overline{p_\ell p_{r+1}}$ crosses $\overline{p_x p_{\ell+1}}$ if and only if it crosses $\overline{q_\ell p_{\ell+1}}$.

Proof. Suppose that $\overline{p_\ell p_{r+1}}$ crosses $\overline{p_x p_{\ell+1}}$. We prove that $\overline{p_\ell p_{r+1}}$ crosses $\overline{q_\ell p_{\ell+1}}$, as well. The proof for the opposite direction is analogous. Let r_h be the intersection point of $\overline{p_x p_{\ell+1}}$ with h_ℓ . Since $\overline{p_x p_{\ell+1}}$ belongs to R , we have that r_h lies to the right of p_ℓ . This implies that, by rotating a ray $\rho(p_\ell, p_{\ell+1})$ starting from p_ℓ and passing through $p_{\ell+1}$ in clockwise direction, around p_ℓ , the point p_{r+1} is encountered before r_h . It follows that, by rotating $\rho(p_\ell, p_{\ell+1})$ in clockwise direction around p_ℓ , the point p_{r+1} is encountered before q_ℓ , as well, since the ray starting at p_ℓ and passing through q_ℓ is the same as the ray starting at p_ℓ and passing through r_h . Hence, $\overline{p_\ell p_{r+1}}$ crosses $\overline{q_\ell p_{\ell+1}}$. $\square$

A corollary of [Claim A.1.12](#) is that the segment $\overline{p_\ell p_{r+1}}$ crosses a bipath B on S_i composed of two monotone st -paths ending at points p_ℓ and p_r , with $\ell < r$, if and only if it crosses any other bipath B' on S_i composed of two monotone st -paths ending at points p_ℓ and p_r . This is obvious if the crossing involves a segment $\overline{p_y p_{y+1}}$, for some $y \in \{\ell + 1, \ell + 2, \dots, r - 1\}$, as such a segment belongs both to B and to B' , whereas it comes from [Claim A.1.12](#) if the crossing involves a segment $\overline{p_x p_{\ell+1}}$ of B or B' with $x < \ell$.

We are now ready to describe the table C and its computation in greater detail. The table C is indexed by triples $\langle p_\ell, p_r, X \rangle$, where p_ℓ and p_r are distinct points in S and $X \in \{L, R\}$. Note that C has $\mathcal{O}(n^2)$ entries. Let $i = \max\{\ell, r\}$. The entry $C[p_\ell, p_r, L]$ is **True** if and only if the segment $\overline{p_\ell p_{i+1}}$ does not cross any bipath B on S_i composed of two monotone *st*-paths L and R ending at points p_ℓ and p_r , if such a bipath exists, otherwise the value of $C[p_\ell, p_r, L]$ is irrelevant. Likewise, the entry $C[p_\ell, p_r, R]$ is **True** if and only if the segment $\overline{p_r p_{i+1}}$ does not cross any bipath B on S_i composed of two monotone *st*-paths L and R ending at points p_ℓ and p_r , if such a bipath exists, otherwise the value of $C[p_\ell, p_r, R]$ is irrelevant.

We show how to compute the entries $C[p_\ell, p_r, L]$, the computation of the entries $C[p_\ell, p_r, R]$ is done analogously. As discussed before, if $\ell > r$, then $C[p_\ell, p_r, L] = \text{True}$; this condition can be verified in $\mathcal{O}(1)$ time, hence in $\mathcal{O}(n^2)$ time over all entries of C . Assume now that $\ell < r = i$. A simple way of computing $C[p_\ell, p_r, L]$ would consist of verifying whether $\overline{p_\ell p_{r+1}}$ intersects any of the segments $\overline{q_\ell p_{\ell+1}}, \overline{p_{\ell+1} p_{\ell+2}}, \dots, \overline{p_{r-1} p_r}$. However, this would take $\Omega(r - \ell)$ time per entry, which would sum up to $\Omega(n^3)$ over all entries of C . Instead, for each fixed $\ell \in [n - 2]$, we compute all the entries $C[p_\ell, p_r, L]$ with $r = \ell + 1, \ell + 2, \dots, n - 1$ in overall $\mathcal{O}(n)$ time, as described below. This sums up to $\mathcal{O}(n^2)$ time over all the entries $C[p_\ell, p_r, L]$ of C with $\ell = 1, 2, \dots, n - 2$ and $r = \ell + 1, \ell + 2, \dots, n - 1$.

Initialize a value α to the angle that is defined by a counter-clockwise rotation around p_ℓ of a horizontal ray starting at p_ℓ and directed rightward, so that the rotation stops when the ray passes through $p_{\ell+1}$. We now look at the values $j = \ell + 1, \ell + 2, \dots, n - 1$ one by one. When we look at a value j , we compute the angle α_j that is defined by a counter-clockwise rotation around p_ℓ of a horizontal ray starting at p_ℓ and directed rightward, so that the rotation stops when the ray passes through p_{j+1} . Two cases can happen. If $\alpha_j \leq \alpha$, as in [Figure A.1.13a](#), then we leave α unaltered and we set $C[p_\ell, p_j, L] = \text{False}$. Otherwise, that is, if $\alpha_j > \alpha$, as in [Figure A.1.13b](#), then we set α to the value of α_j and we set $C[p_\ell, p_j, L] = \text{True}$.

Clearly, this computation takes $\mathcal{O}(1)$ per value j , hence $\mathcal{O}(n)$ time for all the entries $C[p_\ell, p_r, L]$ with $r = \ell + 1, \ell + 2, \dots, n - 1$, and thus $\mathcal{O}(n^2)$ time over all the entries $C[p_\ell, p_r, L]$ of C . Concerning the correctness of the computed values, let $q_1, \dots, q_n$ be n points such that, for $i = 1, \dots, n$, the point q_i has the same y -coordinate as p_i and lies to the right of every point p_j with $j = 1, \dots, n$. It suffices to observe that the straight-line



Figure A.1.13: Computation of the value of the entry $C[p_\ell, p_j, L]$. In (a) we have $\alpha_j \leq \alpha$, hence $C[p_\ell, p_j, L] = \text{False}$, while in (b) we have $\alpha_j > \alpha$, hence $C[p_\ell, p_j, L] = \text{True}$.

segment $\overline{p_\ell p_{j+1}}$ does not cross the polyline $(q_\ell, p_{\ell+1}, p_{\ell+2}, \dots, p_j)$ if and only if a counter-clockwise rotation around p_ℓ of a horizontal ray starting at p_ℓ and directed rightward passes through all of $q_\ell, p_{\ell+1}, p_{\ell+2}, \dots, p_j$ before passing through p_{j+1} . This is expressed by the condition $\alpha_j > \alpha$. As discussed before, assuming that a bipath B on S_j composed of two monotone st -paths ending at p_ℓ and p_j exists, the straight-line segment $\overline{p_\ell p_{j+1}}$ crosses B if and only if it crosses the polyline $(q_\ell, p_{\ell+1}, p_{\ell+2}, \dots, p_j)$, from which the correctness of the computed entry values follows.

We now turn our attention to the computation of the table D , which has $\mathcal{O}(n^2)$ size and allows us to test in $\mathcal{O}(1)$ time whether a bipath B on S_i , with $i \in \{2, \dots, n-1\}$, is safe.

The table D is indexed by triples $\langle p_a, p_b, X \rangle$, where $p_a, p_b \in S$ with $a < b$ and $X \in \{L, R\}$. Each entry of D contains a Boolean value $D[p_a, p_b, X]$ defined as follows.

- Suppose that $X = R$. Consider the rightmost ray $\rho(p_a, S_b \setminus S_a)$ starting from p_a and passing through a point in $S_b \setminus S_a$. We denote by $\mathcal{R}(p_a, p_b)$ the open region of the plane strictly to the right of $\rho(p_a, S_b \setminus S_a)$ and strictly above the horizontal line through p_b . Then, $D[p_a, p_b, R] = \text{True}$ if and only if $\mathcal{R}(p_a, p_b) \cap S \neq \emptyset$.
- Suppose now that $X = L$. Consider the leftmost ray $\ell(p_a, S_b \setminus S_a)$ starting from p_a and passing through a point in $S_b \setminus S_a$. We denote by $\mathcal{L}(p_a, p_b)$ the open region of the plane strictly to the left of the ray $\ell(p_a, S_b \setminus S_a)$ and strictly above the horizontal line passing through p_b . Then, $D[p_a, p_b, L] = \text{True}$ if and only if $\mathcal{L}(p_a, p_b) \cap S \neq \emptyset$.

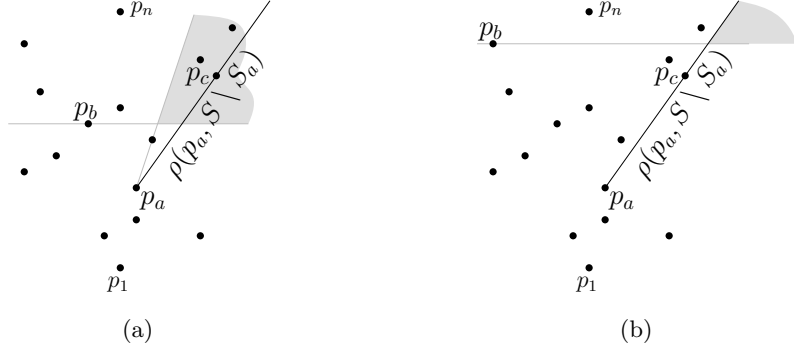


Figure A.1.14: (a) For any $b \in \{a+1, a+2, \dots, c-1\}$, we have that p_c is in $\mathcal{R}(p_a, p_b)$. (b) For any $b \in \{c, c+1, \dots, n\}$, we have that $\mathcal{R}(p_a, p_b)$ is empty. Region $\mathcal{R}(p_a, p_b)$ is shaded gray.

For each fixed $a \in [n-1]$, we show how to compute all the entries $D[p_a, p_b, R]$ with $b = a+1, a+2, \dots, n$ in overall $\mathcal{O}(n)$ time. This sums up to $\mathcal{O}(n^2)$ time over all the entries $D[p_a, p_b, R]$ of D with $a = 1, 2, \dots, n-1$ and $b = a+1, a+2, \dots, n$. The computation of the entries $D[p_a, p_b, L]$ of D is done symmetrically.

We compute the point p_c with $c > a$ such that the ray $\rho(p_a, p_c) = \rho(p_a, S \setminus S_a)$ is the rightmost among the rays starting from p_a and passing through a point in $S \setminus S_a$. This can be done in $\mathcal{O}(n)$ time by inspecting the points $p_{a+1}, p_{a+2}, \dots, p_n$. Then, we set $D[p_a, p_b, R] = \text{True}$ for all the points p_b with $b = a+1, a+2, \dots, c-1$ and $D[p_a, p_b, R] = \text{False}$ for all the points p_b with $b = c, c+1, \dots, n$. Indeed, for any $b \in \{a+1, a+2, \dots, c-1\}$, we have that p_c is in $\mathcal{R}(p_a, p_b)$, since it is strictly above the horizontal line through p_b (given that $b < c$) and strictly to the right of the ray $\rho(p_a, p_b)$ (given that $\rho(p_a, p_c)$ is the rightmost among the rays starting from p_a and passing through a point in $S \setminus S_a$); see [Figure A.1.14a](#). Also, for any $b \in \{c, c+1, \dots, n\}$, we have that p_c is in $S_b \setminus S_a$, and, by definition of p_c , no point is strictly to the right of the ray $\rho(p_a, p_c)$, hence $\mathcal{R}(p_a, p_b)$ is empty; see [Figure A.1.14b](#).

This concludes the description of the $\mathcal{O}(n^2)$ -time computation of the tables C and D . Due to these tables, the computation performed by the enumeration algorithm at each node of the search tree T takes $\mathcal{O}(1)$ time. Indeed, consider a node μ of T associated to a safe bipath B composed of two monotone st -paths

ending at the points p_ℓ and p_r . Let $i = \max\{\ell, r\}$. By means of the value $C[p_\ell, p_r, L]$ and $D[p_r, p_{i+1}, R]$, we can respectively test in $\mathcal{O}(1)$ time whether $B' := B \cup \overline{p_\ell p_{i+1}}$ is a bipath and, in case it is, whether it is safe. If B' is a safe bipath, then the algorithm adds to μ a child corresponding to B' , and the traversal continues on that child. Once the traversal backtracks to μ again, or if B' was not a safe bipath in the first place, by means of the values $C[p_\ell, p_r, R]$ and $D[p_\ell, p_{i+1}, L]$, we can respectively test in $\mathcal{O}(1)$ time whether $B'' := B \cup \overline{p_r p_{i+1}}$ is a bipath and, in case it is, whether it is safe. If B'' is a safe bipath, then the algorithm adds to μ a child corresponding to B'' , and the traversal continues on that child. Since the computation at each node takes $\mathcal{O}(1)$ time and since T has n levels, it follows that the algorithm's delay is in $\mathcal{O}(n)$.

Items **i** to **v** complete the proof of [Theorem A.1.10](#).

A.1.7 Conclusions and Open Problems

We addressed basic pointset embeddability problems for upward planar graphs. We proved that UPSE testing is NP-hard even for planar st -graphs composed of internally-disjoint st -paths and for directed trees composed of directed root-to-leaf paths. For planar st -graphs, we showed that UPSE TESTING can be solved in $\mathcal{O}(n^{4k})$ time, where k is the maximum st -cutset of G , and we provided an algorithm to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay. We also showed how to enumerate all monotone polygonalizations of a given pointset with $\mathcal{O}(n)$ worst-case delay. We point out the following open problems.

- Our NP-hardness proofs for UPSE TESTING use the fact that the points are not in general position. Given a directed tree T on n vertices and a set S of n points *in general position*, is it NP-hard to decide whether T has an UPSE on S ?
- Can UPSE TESTING be solved in polynomial time or does it remain NP-hard if the input is a *maximal* planar st -graph?
- We proved that UPSE TESTING for a planar st -graph is in XP with respect to the size of the maximum st -cutset of G . Is the problem in FPT with respect to the same parameter? Are there other interesting parameterizations for the problem?
- Let S be a pointset and $\mathcal{P}$ be a non-crossing path on a subset of S . Is it possible to decide in polynomial time whether $\mathcal{P}$ can be extended to a polygonalization of S ? A positive answer would imply an algorithm with

polynomial delay for enumerating the polygonalizations of a pointset, with the same approach as the one we adopted in this chapter for monotone polygonalizations.

Appendix B

Publications

International Journal Publications

- [J1] Carlos Alegría, Susanna Caroppo, Giordano Da Lozzo, **Marco D’Elia**, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, Maurizio Patrignani. Upward Pointset Embeddings of Planar st-Graphs. *Algorithmica*, 87(6): 930-960, 2025. doi:[10.1007/S00453-025-01302-2](https://doi.org/10.1007/S00453-025-01302-2)
- [J2] **Marco D’Elia**, Irene Finocchi, Maurizio Patrignani. Maximal cliques summarization: Principles, problem classification, and algorithmic approaches. *Computer Science Review*, 58(100784), 2025. doi:[10.1016/J.COSREV.2025.100784](https://doi.org/10.1016/J.COSREV.2025.100784)

International Conference Publications

- [C1] **Marco D’Elia**, Fabrizio Frati. Large Induced Subgraphs of Bounded Degree in Outerplanar and Planar Graphs. The 20th International Conference and Workshops on Algorithms and Computation (WALCOM): 187-202, 2026. doi:[10.1007/978-981-95-7127-7_13](https://doi.org/10.1007/978-981-95-7127-7_13)
- [C2] **Marco D’Elia**, Irene Finocchi, Maurizio Patrignani. Engineering Algorithms for ℓ -Isolated Maximal Clique Enumeration. The 20th International Conference and Workshops on Algorithms and Computation (WALCOM): 545-559, 2026. doi:[10.1007/978-981-95-7127-7_36](https://doi.org/10.1007/978-981-95-7127-7_36)

B. PUBLICATIONS

- [C3] Carlos Alegría, Susanna Caroppo, Giordano Da Lozzo, **Marco D’Elia**, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, Maurizio Patrignani. Upward Pointset Embeddings of Planar st-Graphs. 32nd International Symposium on Graph Drawing and Network Visualization (GD): 24:1-24:18, 2024. doi:[10.4230/LIPICS.GD.2024.24](https://doi.org/10.4230/LIPICS.GD.2024.24)
- [C4] **Marco D’Elia**, Irene Finocchi, Maurizio Patrignani. Clique-TF-IDF: A New Partitioning Framework Based on Dense Substructures. Advances in Artificial Intelligence (AI*IA): 396-410, 2023. doi:[10.1007/978-3-031-47546-7_27](https://doi.org/10.1007/978-3-031-47546-7_27)

Technical Reports

- [T1] **Marco D’Elia**, Fabrizio Frati. On Large Induced Outerplanar Subgraphs in 2-Outerplanar Graphs. CoRR, vol. abs/2602.18121, 2026. doi:[10.48550/ARXIV.2602.18121](https://doi.org/10.48550/ARXIV.2602.18121)
- [T2] **Marco D’Elia**, Irene Finocchi, Maurizio Patrignani. High-Modularity Graph Partitioning Through NLP Techniques and Maximal Clique Enumeration. CoRR, vol. abs/2602.23948, 2026. doi:[10.48550/ARXIV.2602.23948](https://doi.org/10.48550/ARXIV.2602.23948)
- [T3] **Marco D’Elia**, Irene Finocchi, Maurizio Patrignani. Engineering Algorithms for ℓ -Isolated Maximal Clique Enumeration. CoRR, vol. abs/2511.03525, 2025. doi:[10.48550/ARXIV.2511.03525](https://doi.org/10.48550/ARXIV.2511.03525)
- [T4] **Marco D’Elia**, Fabrizio Frati. Large Induced Subgraphs of Bounded Degree in Outerplanar and Planar Graphs. CoRR, vol. abs/2412.14784, 2024. doi:[10.48550/ARXIV.2412.14784](https://doi.org/10.48550/ARXIV.2412.14784)
- [T5] Carlos Alegría, Susanna Caroppo, Giordano Da Lozzo, **Marco D’Elia**, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, Maurizio Patrignani. Upward Pointset Embeddings of Planar st-Graphs. CoRR, vol. abs/2408.17369, 2024. doi:[10.48550/ARXIV.2408.17369](https://doi.org/10.48550/ARXIV.2408.17369)

Bibliography

- [1] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Miller, J. W. Thatcher (Eds.), Proc. Symp. Compl. Comp. Computat., Plenum Press, New York, 1972, pp. 85–103. doi:[10.1007/978-1-4684-2001-2_9](https://doi.org/10.1007/978-1-4684-2001-2_9).
- [2] R. G. Downey, M. R. Fellows, Fixed-parameter tractability and completeness II: on completeness for W[1], Theor. Comput. Sci. 141 (1&2) (1995) 109–131. doi:[10.1016/0304-3975\(94\)00097-3](https://doi.org/10.1016/0304-3975(94)00097-3).
- [3] R. G. Downey, M. R. Fellows, Parameterized Complexity, Monographs in Computer Science, Springer, 1999. doi:[10.1007/978-1-4612-0515-9](https://doi.org/10.1007/978-1-4612-0515-9).
- [4] J. Hastad, Clique is hard to approximate within $n^{1-\epsilon}$, Acta Mathematica 182 (1) (1999) 105–142. doi:[10.1007/BF02392825](https://doi.org/10.1007/BF02392825).
- [5] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, in: J. M. Kleinberg (Ed.), Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21–23, 2006, ACM, 2006, pp. 681–690. doi:[10.1145/1132516.1132612](https://doi.org/10.1145/1132516.1132612).
- [6] C. Bron, J. Kerbosch, Finding all cliques of an undirected graph (algorithm 457), Commun. ACM 16 (9) (1973) 575–576. doi:[10.1145/362342.362367](https://doi.org/10.1145/362342.362367).

BIBLIOGRAPHY

- [7] F. Cazals, C. Karande, A note on the problem of reporting maximal cliques, *Theor. Comput. Sci.* 407 (1-3) (2008) 564–568. doi:[10.1016/j.tcs.2008.05.010](https://doi.org/10.1016/j.tcs.2008.05.010).
- [8] D. Eppstein, M. Löffler, D. Strash, Listing all maximal cliques in large sparse real-world graphs, *ACM J. Exp. Algorithmics* 18 (2013). doi:[10.1145/2543629](https://doi.org/10.1145/2543629).
- [9] I. Koch, Enumerating all connected maximal common subgraphs in two graphs, *Theor. Comput. Sci.* 250 (1-2) (2001) 1–30. doi:[10.1016/S0304-3975\(00\)00286-3](https://doi.org/10.1016/S0304-3975(00)00286-3).
- [10] E. Tomita, A. Tanaka, H. Takahashi, The worst-case time complexity for generating all maximal cliques, in: K. Chwa, J. I. Munro (Eds.), *COCOON 2004*, Vol. 3106, Springer, Germany, 2004, pp. 161–170. doi:[10.1007/978-3-540-27798-9_19](https://doi.org/10.1007/978-3-540-27798-9_19).
- [11] Y. Jin, B. Xiong, K. He, Y. Zhou, Y. Zhou, On fast enumeration of maximal cliques in large graphs, *Expert Syst. Appl.* 187 (2022) 115915. doi:[10.1016/J.ESWA.2021.115915](https://doi.org/10.1016/J.ESWA.2021.115915).
- [12] A. Conte, R. Grossi, A. Marino, L. Versari, Sublinear-space and bounded-delay algorithms for maximal clique enumeration in graphs, *Algorithmica* 82 (6) (2020) 1547–1573. doi:[10.1007/S00453-019-00656-8](https://doi.org/10.1007/S00453-019-00656-8).
- [13] J. Moon, L. Moser, On cliques in graphs, *Israel Journal of Mathematics* 3 (1965) 23–28. doi:[10.1007/BF02760024](https://doi.org/10.1007/BF02760024).
- [14] G. Di Battista, P. Eades, R. Tamassia, I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall, 1999.
- [15] G. Borradaile, H. Le, M. Sherman-Bennett, Large induced acyclic and outerplanar subgraphs of 2-outerplanar graph, *Graphs Comb.* 33 (6) (2017) 1621–1634. doi:[10.1007/S00373-017-1859-3](https://doi.org/10.1007/S00373-017-1859-3).
- [16] J. R. Gilbert, J. P. Hutchinson, R. E. Tarjan, A separator theorem for graphs of bounded genus, *J. Algorithms* 5 (3) (1984) 391–407. doi:[10.1016/0196-6774\(84\)90019-1](https://doi.org/10.1016/0196-6774(84)90019-1).
- [17] K. Kawarabayashi, B. A. Reed, A separator theorem in minor-closed classes, in: *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010*, Las Vegas, Nevada, USA, October 23-26, 2010, IEEE Computer Society, 2010, pp. 153–162. doi:[10.1109/FOCS.2010.22](https://doi.org/10.1109/FOCS.2010.22).

BIBLIOGRAPHY

- [18] R. J. Lipton, R. E. Tarjan, Applications of a planar separator theorem, *SIAM J. Comput.* 9 (3) (1980) 615–627. doi:[10.1137/0209046](https://doi.org/10.1137/0209046).
- [19] B. S. Baker, Approximation algorithms for NP-complete problems on planar graphs, *J. ACM* 41 (1) (1994) 153–180. doi:[10.1145/174644.174650](https://doi.org/10.1145/174644.174650).
- [20] E. D. Demaine, M. Hajiaghayi, B. Mohar, Approximation algorithms via contraction decomposition, *Comb.* 30 (5) (2010) 533–552. doi:[10.1007/S00493-010-2341-5](https://doi.org/10.1007/S00493-010-2341-5).
- [21] M. O. Albertson, D. M. Berman, A conjecture on planar graphs, *Graph theory and related topics* (1979) 357.
- [22] J. Akiyama, M. Watanabe, Maximum induced forests of planar graphs, *Graphs Comb.* 3 (1) (1987) 201–202. doi:[10.1007/BF01788541](https://doi.org/10.1007/BF01788541).
- [23] M. J. Pelsmayer, Maximum induced linear forests in outerplanar graphs, *Graphs Comb.* 20 (1) (2004) 121–129. doi:[10.1007/S00373-003-0528-X](https://doi.org/10.1007/S00373-003-0528-X).
- [24] E. Tomita, A. Tanaka, H. Takahashi, The worst-case time complexity for generating all maximal cliques and computational experiments, *Theor. Comput. Sci.* 363 (1) (2006) 28–42. doi:[10.1016/J.TCS.2006.06.015](https://doi.org/10.1016/J.TCS.2006.06.015).
- [25] R. Diestel, *Graph Theory*, 4th Edition, Vol. 173 of Graduate texts in mathematics, Springer, 2012.
- [26] J. Cheng, Y. Ke, A. W.-C. Fu, J. X. Yu, L. Zhu, Finding maximal cliques in massive networks, *ACM Trans. Database Syst.* 36 (4) (2011) 21. doi:[10.1145/2043652.2043654](https://doi.org/10.1145/2043652.2043654).
- [27] J. Cheng, L. Zhu, Y. Ke, S. Chu, Fast algorithms for maximal clique enumeration with limited memory, in: *KDD*, 2012, pp. 1240–1248. doi:[10.1145/2339530.2339724](https://doi.org/10.1145/2339530.2339724).
- [28] A. Conte, R. De Virgilio, A. Maccioni, M. Patrignani, R. Torlone, Finding all maximal cliques in very large social networks, in: *EDBT 2016*, OpenProceedings.org, Konstanz, Germany, 2016, pp. 173–184. doi:[10.5441/002/edbt.2016.18](https://doi.org/10.5441/002/edbt.2016.18).

BIBLIOGRAPHY

- [29] Y. Xu, J. Cheng, A. W. Fu, Distributed maximal clique computation and management, *IEEE Trans. Serv. Comput.* 9 (1) (2016) 110–122. doi:[10.1109/TSC.2015.2479225](https://doi.org/10.1109/TSC.2015.2479225).
- [30] A. Conte, E. Tomita, On the overall and delay complexity of the CLIQUES and Bron-Kerbosch algorithms, *Theor. Comput. Sci.* 899 (C) (2022) 1–24. doi:[10.1016/j.tcs.2021.11.005](https://doi.org/10.1016/j.tcs.2021.11.005).
- [31] J. Fox, T. Roughgarden, C. Seshadhri, F. Wei, N. Wein, Finding cliques in social networks: A new distribution-free model, in: I. Chatzigiannakis, C. Kaklamanis, D. Marx, D. Sannella (Eds.), 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9–13, 2018, Prague, Czech Republic, Vol. 107 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 55:1–55:15. doi:[10.4230/LIPICS.ICALP.2018.55](https://doi.org/10.4230/LIPICS.ICALP.2018.55).
- [32] J. Fox, T. Roughgarden, C. Seshadhri, F. Wei, N. Wein, Finding cliques in social networks: A new distribution-free model, *SIAM J. Comput.* 49 (2) (2020) 448–464. doi:[10.1137/18M1210459](https://doi.org/10.1137/18M1210459).
- [33] S. B. Seidman, B. L. Foster, A graph-theoretic generalization of the clique concept*, *The Journal of Mathematical Sociology* 6 (1) (1978) 139–154. doi:[10.1080/0022250X.1978.9989883](https://doi.org/10.1080/0022250X.1978.9989883).
- [34] A. Conte, D. Firmani, C. Mordente, M. Patrignani, R. Torlone, Fast enumeration of large k-plexes, in: T. Eliassi-Rad, R. Kumar (Eds.), 23rd SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2017), ACM, 2017, pp. 115–124. doi:<http://dx.doi.org/10.1145/3097983.3098031>.
- [35] A. Conte, D. Firmani, M. Patrignani, R. Torlone, A meta algorithm for finding large k-plexes, *Knowledge and Information Systems* 63 (2021) 1745–1769. doi:<https://doi.org/10.1007/s10115-021-01570-8>.
- [36] A. Conte, D. Firmani, M. Patrignani, R. Torlone, Shared-nothing distributed enumeration of 2-plexes, in: P. Cui, E. Rundensteiner, D. Carmel, Q. He, J. Yu (Eds.), 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), ACM, 2019, pp. 2469–2472. doi:<https://doi.org/10.1145/3357384.3358083>.
- [37] H. Yu, A. Paccanaro, V. Trifonov, M. Gerstein, Predicting interactions in protein networks by completing defective cliques, *Bioinform.* 22 (7) (2006) 823–829. doi:[10.1093/BIOINFORMATICS/BTL014](https://doi.org/10.1093/BIOINFORMATICS/BTL014).

BIBLIOGRAPHY

- [38] J. Abello, M. G. C. Resende, S. Sudarsky, Massive quasi-clique detection, in: S. Rajsbaum (Ed.), LATIN 2002: Theoretical Informatics, 5th Latin American Symposium, Cancun, Mexico, April 3-6, 2002, Proceedings, Vol. 2286 of Lecture Notes in Computer Science, Springer, 2002, pp. 598–612. [doi:10.1007/3-540-45995-2_51](https://doi.org/10.1007/3-540-45995-2_51).
- [39] J. Pattillo, N. Youssef, S. Butenko, Clique relaxation models in social network analysis, in: M. T. Thai, P. M. Pardalos (Eds.), Handbook of Optimization in Complex Networks: Communication and Social Networks, Springer New York, New York, NY, 2012, pp. 143–162. [doi:10.1007/978-1-4614-0857-4_5](https://doi.org/10.1007/978-1-4614-0857-4_5).
- [40] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, D. Wagner, On finding graph clusterings with maximum modularity, in: WG 2007, Vol. 4769, Springer, Germany, 2007, pp. 121–132. [doi:10.1007/978-3-540-74839-7_12](https://doi.org/10.1007/978-3-540-74839-7_12).
- [41] M. D’Elia, I. Finocchi, M. Patrignani, Clique-tf-idf: A new partitioning framework based on dense substructures, in: R. Basili, D. Lembo, C. Limongelli, A. Orlandini (Eds.), AIXIA 2023, Vol. 14318 of Lecture Notes in Computer Science, Springer, 2023, pp. 396–410. [doi:10.1007/978-3-031-47546-7_27](https://doi.org/10.1007/978-3-031-47546-7_27).
- [42] M. D’Elia, I. Finocchi, M. Patrignani, High-modularity graph partitioning through NLP techniques and maximal clique enumeration, CoRR abs/2602.23948 (2026). [doi:10.48550/ARXIV.2602.23948](https://doi.org/10.48550/ARXIV.2602.23948).
- [43] P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar, Introduction to Data Mining (2nd Edition), 2nd Edition, Pearson, , 2018.
- [44] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, Oakland, CA, USA, 1967, pp. 281–297.
- [45] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1998) 359–392. [doi:10.1137/S1064827595287997](https://doi.org/10.1137/S1064827595287997).
- [46] V. A. Traag, L. Waltman, N. J. Van Eck, From louvain to leiden: guaranteeing well-connected communities, Scientific reports 9 (1) (2019) 1–12. [doi:10.1038/s41598-019-41695-z](https://doi.org/10.1038/s41598-019-41695-z).

BIBLIOGRAPHY

- [47] F. Devvrit, A. Sinha, I. S. Dhillon, P. Jain, S3GC: scalable self-supervised graph clustering, in: NeurIPS, 2022.
- [48] D. Combe, C. Largeron, M. Géry, E. Egyed-Zsigmond, I-louvain: An attributed graph clustering method, in: É. Fromont, T. D. Bie, M. van Leeuwen (Eds.), IDA 2015, Vol. 9385 of Lecture Notes in Computer Science, Springer, Germany, 2015, pp. 181–192. doi:[10.1007/978-3-319-24465-5_16](https://doi.org/10.1007/978-3-319-24465-5_16).
- [49] N. Bansal, A. Blum, S. Chawla, Correlation clustering, Mach. Learn. 56 (1-3) (2004) 89–113. doi:[10.1023/B:MACH.0000033116.57574.95](https://doi.org/10.1023/B:MACH.0000033116.57574.95).
- [50] B. Saha, S. Subramanian, Correlation clustering with same-cluster queries bounded by optimal cost, in: M. A. Bender, O. Svensson, G. Herman (Eds.), ESA 2019, Vol. 144 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2019, pp. 81:1–81:17. doi:[10.4230/LIPIcs.ESA.2019.81](https://doi.org/10.4230/LIPIcs.ESA.2019.81).
- [51] S. Lattanzi, B. Moseley, S. Vassilvitskii, Y. Wang, R. Zhou, Robust online correlation clustering, in: M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, J. W. Vaughan (Eds.), NeurIPS 2021, 2021, pp. 4688–4698.
- [52] Y. Chen, J. You, J. He, Y. Lin, Y. Peng, C. Wu, Y. Zhu, SP-GNN: learning structure and position information from graphs, Neural Networks 161 (2023) 505–514. doi:[10.1016/j.neunet.2023.01.051](https://doi.org/10.1016/j.neunet.2023.01.051).
- [53] H. Cui, Z. Lu, P. Li, C. Yang, On positional and structural node features for graph neural networks on non-attributed graphs, in: M. A. Hasan, L. Xiong (Eds.), ACM CIKM 2022, ACM, USA, 2022, pp. 3898–3902. doi:[10.1145/3511808.3557661](https://doi.org/10.1145/3511808.3557661).
- [54] J. Zhu, X. Lu, M. Heimann, D. Koutra, Node proximity is all you need: Unified structural and positional node and graph embedding, in: C. Demeniconi, I. Davidson (Eds.), SIAM Int. Conf. on Data Mining, SDM 2021, SIAM, USA, 2021, pp. 163–171. doi:[10.1137/1.9781611976700.19](https://doi.org/10.1137/1.9781611976700.19).
- [55] L. F. R. Ribeiro, P. H. P. Saverese, D. R. Figueiredo, *struc2vec*: Learning node representations from structural identity, in: ACM SIGKDD 2017, ACM, USA, 2017, pp. 385–394. doi:[10.1145/3097983.3098061](https://doi.org/10.1145/3097983.3098061).
- [56] E. Al-sharoha, B. Rahahleh, Community detection in networks through a deep robust auto-encoder nonnegative matrix factorization, Engineering

BIBLIOGRAPHY

- Applications of Artificial Intelligence 118 (2023) 105657. doi:<https://doi.org/10.1016/j.engappai.2022.105657>.
- [57] A. Clauset, M. E. J. Newman, C. Moore, Finding community structure in very large networks, *Phys. Rev. E* 70 (2004) 066111. doi:[10.1103/PhysRevE.70.066111](https://doi.org/10.1103/PhysRevE.70.066111).
- [58] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment* 2008 (10) (2008) P10008. doi:[10.1088/1742-5468/2008/10/p10008](https://doi.org/10.1088/1742-5468/2008/10/p10008).
- [59] P. Pons, M. Latapy, Computing communities in large networks using random walks (long version) (2005). doi:[10.48550/ARXIV.PHYSICS/0512106](https://doi.org/10.48550/ARXIV.PHYSICS/0512106).
- [60] M. Rosvall, C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, *Proceedings of the National Academy of Sciences* 105 (4) (2008) 1118–1123. doi:[10.1073/pnas.0706851105](https://doi.org/10.1073/pnas.0706851105).
- [61] G. Cordasco, L. Gargano, Community detection via semi-synchronous label propagation algorithms, in: 2010 IEEE International Workshop on: Business Applications of Social Network Analysis (BASNA), 2010, pp. 1–8. doi:[10.1109/BASNA.2010.5730298](https://doi.org/10.1109/BASNA.2010.5730298).
- [62] J. Reichardt, S. Bornholdt, Statistical mechanics of community detection, *Physical Review E* 74 (1) (jul 2006). doi:[10.1103/physreve.74.016110](https://doi.org/10.1103/physreve.74.016110).
- [63] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, *Proceedings of the National Academy of Sciences* 99 (12) (2002) 7821–7826. doi:[10.1073/pnas.122653799](https://doi.org/10.1073/pnas.122653799).
- [64] C. Pizzuti, Ga-net: A genetic algorithm for community detection in social networks, in: *Proc. of the 10th Int. Conf. on Parallel Problem Solving from Nature (PPSN X)*, Vol. 5199, Springer-Verlag, 2008, p. 1081–1090. doi:[10.1007/978-3-540-87700-4_107](https://doi.org/10.1007/978-3-540-87700-4_107).
- [65] A. Prat-Pérez, D. Dominguez-Sal, J.-L. Larriba-Pey, High quality, scalable and parallel community detection for large real graphs, in: *Proc. WWW 2014*, Association for Computing Machinery, New York, NY, USA, 2014, p. 225–236. doi:[10.1145/2566486.2568010](https://doi.org/10.1145/2566486.2568010).

BIBLIOGRAPHY

- [66] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, R. Rastogi (Eds.), ACM SIGKDD 2016, ACM, USA, 2016, pp. 855–864. [doi:10.1145/2939672.2939754](https://doi.org/10.1145/2939672.2939754).
- [67] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, R. Ghani (Eds.), The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, ACM, USA, 2014, pp. 701–710. [doi:10.1145/2623330.2623732](https://doi.org/10.1145/2623330.2623732).
- [68] X. Liu, J. Tang, Network representation learning: A macro and micro view, *AI Open* 2 (2021) 43–64. [doi:10.1016/J.AIOPEN.2021.02.001](https://doi.org/10.1016/J.AIOPEN.2021.02.001).
- [69] S. Zhao, J. Chen, J. Chen, Y. Zhang, J. Tang, Hierarchical label with imbalance and attributed network structure fusion for network embedding, *AI Open* 3 (2022) 91–100. [doi:10.1016/J.AIOPEN.2022.07.002](https://doi.org/10.1016/J.AIOPEN.2022.07.002).
- [70] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick, On the permanence of vertices in network communities, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 1396–1405. [doi:10.1145/2623330.2623707](https://doi.org/10.1145/2623330.2623707).
- [71] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, UK, 2008.
- [72] J. Leskovec, A. Krevl, SNAP Datasets: Stanford large network dataset collection, <http://snap.stanford.edu/data> (Jun. 2014).
- [73] J. Kunegis, KONECT: the koblenz network collection, in: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume, ACM, 2013, pp. 1343–1350. [doi:10.1145/2487788.2488173](https://doi.org/10.1145/2487788.2488173).
- [74] R. A. Rossi, N. K. Ahmed, [The network data repository with interactive graph analytics and visualization](https://networkrepository.com), in: AAAI, 2015. URL <https://networkrepository.com>
- [75] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, *Phys. Rev. E* 78 (2008) 046110. [doi:10.1103/PhysRevE.78.046110](https://doi.org/10.1103/PhysRevE.78.046110).

BIBLIOGRAPHY

- [76] G. W. Flake, S. Lawrence, C. L. Giles, Efficient identification of web communities, in: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 2000, pp. 150–160. [doi:10.1145/347090.347121](#).
- [77] H. Saito, M. Toyoda, M. Kitsuregawa, K. Aihara, A large-scale study of link spam detection by graph algorithms, in: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web, 2007, pp. 45–48. [doi:10.1145/1244408.1244417](#).
- [78] V. Boginski, S. Butenko, P. M. Pardalos, Statistical analysis of financial networks, *Computational statistics & data analysis* 48 (2) (2005) 431–443. [doi:10.1016/J.CSDA.2004.02.004](#).
- [79] V. Boginski, S. Butenko, P. M. Pardalos, Mining market data: A network approach, *Computers & Operations Research* 33 (11) (2006) 3171–3184. [doi:10.1016/J.COR.2005.01.027](#).
- [80] S. Butenko, W. E. Wilhelm, Clique-detection models in computational biochemistry and genomics, *European Journal of Operational Research* 173 (1) (2006) 1–17. [doi:10.1016/J.EJOR.2005.05.026](#).
- [81] E. J. Chesler, L. Lu, S. Shou, Y. Qu, J. Gu, J. Wang, H. C. Hsu, J. D. Mountz, N. E. Baldwin, M. A. Langston, et al., Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function, *Nature genetics* 37 (3) (2005) 233–242. [doi:10.1038/ng1518](#).
- [82] A. Töpfer, T. Marschall, R. A. Bull, F. Luciani, A. Schönhuth, N. Beerenwinkel, Viral quasispecies assembly via maximal clique enumeration, *PLoS computational biology* 10 (3) (2014) e1003515. [doi:10.1371/JOURNAL.PCBI.1003515](#).
- [83] A. Bhar, L. C. Gierse, A. Meene, H. Wang, C. Karte, T. Schwaiger, C. Schröder, T. C. Mettenleiter, T. Urich, K. Riedel, L. Kaderali, Application of a maximal-clique based community detection algorithm to gut microbiome data reveals driver microbes during influenza a virus infection, *Frontiers in Microbiology* 13 (2022). [doi:10.3389/fmicb.2022.979320](#).
- [84] F. Chen, H. Zhai, Y. Fang, Available bandwidth in multirate and multihop wireless ad hoc networks, *IEEE J. Sel. Areas Commun.* 28 (3) (2010) 299–307. [doi:10.1109/JSAC.2010.100402](#).

BIBLIOGRAPHY

- [85] K. Jain, J. Padhye, V. N. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, *Wirel. Networks* 11 (4) (2005) 471–487. doi:[10.1007/S11276-005-1769-9](https://doi.org/10.1007/S11276-005-1769-9).
- [86] B. Balasundaram, S. Butenko, Graph domination, coloring and cliques in telecommunications, in: M. G. C. Resende, P. M. Pardalos (Eds.), *Handbook of Optimization in Telecommunications*, Springer, 2006, pp. 865–890. doi:[10.1007/978-0-387-30165-5_30](https://doi.org/10.1007/978-0-387-30165-5_30).
- [87] J. Wang, J. Cheng, A. W. Fu, Redundancy-aware maximal cliques, in: I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, R. Uthrusamy (Eds.), *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, ACM, 2013, pp. 122–130. doi:[10.1145/2487575.2487689](https://doi.org/10.1145/2487575.2487689).
- [88] F. N. Afrati, D. Fotakis, J. D. Ullman, Enumerating subgraph instances using map-reduce, in: C. S. Jensen, C. M. Jermaine, X. Zhou (Eds.), *29th IEEE International Conference on Data Engineering, ICDE 2013*, IEEE Computer Society, 2013, pp. 62–73. doi:[10.1109/ICDE.2013.6544814](https://doi.org/10.1109/ICDE.2013.6544814).
- [89] I. Finocchi, M. Finocchi, E. G. Fusco, Clique counting in mapreduce: Algorithms and experiments, *ACM J. Exp. Algorithmics* 20 (2015) 1.7:1–1.7:20. doi:[10.1145/2794080](https://doi.org/10.1145/2794080).
- [90] E. Coppa, I. Finocchi, R. L. Garcia, Counting cliques in parallel without a cluster: Engineering a fork/join algorithm for shared-memory platforms, *Inf. Sci.* 496 (2019) 553–571. doi:[10.1016/J.INS.2018.07.018](https://doi.org/10.1016/J.INS.2018.07.018).
- [91] I. Finocchi, R. L. Garcia, B. Sinimeri, From stars to diamonds: Counting and listing almost complete subgraphs in large networks, *Comput. J.* 67 (6) (2024) 2151–2161. doi:[10.1093/COMJNL/BXAD129](https://doi.org/10.1093/COMJNL/BXAD129).
- [92] R. Mokken, Cliques, clubs and clans, *Quality & Quantity* 13 (2) (1979) 161–173. doi:<https://doi.org/10.1007/BF00139635>.
- [93] M. D’Elia, I. Finocchi, M. Patrignani, Maximal cliques summarization: Principles, problem classification, and algorithmic approaches, *Computer Science Review* 58 (2025) 100784. doi:[10.1016/j.cosrev.2025.100784](https://doi.org/10.1016/j.cosrev.2025.100784).
- [94] L. Yuan, L. Qin, X. Lin, L. Chang, W. Zhang, Diversified top-k clique search, *VLDB J.* 25 (2) (2016) 171–196. doi:[10.1007/S00778-015-0408-Z](https://doi.org/10.1007/S00778-015-0408-Z).

BIBLIOGRAPHY

- [95] Y. Uno, Y. Ota, A. Uemichi, Web structure mining by isolated cliques, *IEICE Trans. Inf. Syst.* 90-D (12) (2007) 1998–2006. doi:[10.1093/IETISY/E90-D.12.1998](https://doi.org/10.1093/IETISY/E90-D.12.1998).
- [96] T. Shigezumi, Y. Uno, O. Watanabe, A new model for a scale-free hierarchical structure of isolated cliques, *J. Graph Algorithms Appl.* 15 (5) (2011) 661–682. doi:[10.7155/JGAA.00243](https://doi.org/10.7155/JGAA.00243).
- [97] Y. Uno, F. Oguri, Contracted webgraphs - scale-freeness and structure mining -, *IEICE Trans. Commun.* 96-B (11) (2013) 2766–2773. doi:[10.1587/TRANSCOM.E96.B.2766](https://doi.org/10.1587/TRANSCOM.E96.B.2766).
- [98] H. Ito, K. Iwama, Enumeration of isolated cliques and pseudo-cliques, *ACM Trans. Algorithms* 5 (4) (2009) 40:1–40:21. doi:[10.1145/1597036.1597044](https://doi.org/10.1145/1597036.1597044).
- [99] F. Hüffner, C. Komusiewicz, H. Moser, R. Niedermeier, Isolation concepts for clique enumeration: Comparison and computational experiments, *Theor. Comput. Sci.* 410 (52) (2009) 5384–5397. doi:[10.1016/J.TCS.2009.05.008](https://doi.org/10.1016/J.TCS.2009.05.008).
- [100] C. Komusiewicz, F. Hüffner, H. Moser, R. Niedermeier, Isolation concepts for efficiently enumerating dense subgraphs, *Theor. Comput. Sci.* 410 (38–40) (2009) 3640–3654. doi:[10.1016/J.TCS.2009.04.021](https://doi.org/10.1016/J.TCS.2009.04.021).
- [101] N. Modani, K. Dey, Large maximal cliques enumeration in sparse graphs, in: J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K. Choi, A. Chowdhury (Eds.), *CIKM 2008*, ACM, 2008, pp. 1377–1378. doi:[10.1145/1458082.1458288](https://doi.org/10.1145/1458082.1458288).
- [102] N. Modani, K. Dey, Large maximal cliques enumeration in large sparse graphs, in: S. Chawla, K. Karlapalem, V. Pudi (Eds.), *Proceedings of the 15th International Conference on Management of Data*, December 9–12, 2009, International School of Information Management, Mysore, India, Computer Society of India, 2009.
- [103] J. Bhasker, T. Samad, The clique-partitioning problem, *Computers & Mathematics with Applications* 22 (6) (1991) 1–11. doi:[https://doi.org/10.1016/0898-1221\(91\)90001-K](https://doi.org/10.1016/0898-1221(91)90001-K).
- [104] J. T. Kim, D. R. Shin, New efficient clique partitioning algorithms for register-transfer synthesis of data paths, *Journal-Korean Physical Society* 40 (2002) 754–758.

BIBLIOGRAPHY

- [105] S. Sundar, A. Singh, Two grouping-based metaheuristics for clique partitioning problem, *Appl. Intell.* 47 (2) (2017) 430–442. doi:10.1007/S10489-017-0904-5.
- [106] N. Pullman, H. Shank, W. Wallis, Clique coverings of graphs v: maximal-clique partitions, *Bulletin of the Australian Mathematical Society* 25 (3) (1982) 337–356. doi:10.1017/S0004972700005414.
- [107] I. Holyer, The NP-completeness of some edge-partition problems, *SIAM Journal on Computing* 10 (4) (1981) 713–717. doi:10.1137/0210054.
- [108] P. Erdős, R. Faudree, E. T. Ordman, Clique partitions and clique coverings, *Discrete Mathematics* 72 (1-3) (1988) 93–101. doi:10.1016/0012-365X(88)90197-5.
- [109] N. J. Pullman, Clique covering of graphs iv. algorithms, *SIAM Journal on Computing* 13 (1) (1984) 57–75. doi:10.1137/0213005.
- [110] D. A. Gregory, S. McGuinness, W. Wallis, Clique partitions of the cocktail party graph, *Discrete Mathematics* 59 (3) (1986) 267–273. doi:10.1016/0012-365X(86)90173-1.
- [111] F. Hüffner, C. Komusiewicz, H. Moser, R. Niedermeier, Fixed-parameter algorithms for cluster vertex deletion, *Theory Comput. Syst.* 47 (1) (2010) 196–217. doi:10.1007/S00224-008-9150-X.
- [112] A. Boral, M. Cygan, T. Kociumaka, M. Pilipczuk, A fast branching algorithm for cluster vertex deletion, *Theory Comput. Syst.* 58 (2) (2016) 357–376. doi:10.1007/S00224-015-9631-7.
- [113] Z. Ertem, E. Lykhovyd, Y. Wang, S. Butenko, The maximum independent union of cliques problem: complexity and exact approaches, *J. Glob. Optim.* 76 (3) (2020) 545–562. doi:10.1007/S10898-018-0694-2.
- [114] K. Tian, M. Xiao, B. Yang, Parameterized algorithms for cluster vertex deletion on degree-4 graphs and general graphs, in: W. Wu, G. Tong (Eds.), *Computing and Combinatorics - 29th International Conference, COCOON 2023, Hawaii, HI, USA, December 15-17, 2023, Proceedings, Part I*, Vol. 14422 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 182–194. doi:10.1007/978-3-031-49190-0_13.

BIBLIOGRAPHY

- [115] M. Aprile, M. Drescher, S. Fiorini, T. Huynh, A tight approximation algorithm for the cluster vertex deletion problem, *Math. Program.* 197 (2) (2023) 1069–1091. doi:[10.1007/S10107-021-01744-W](https://doi.org/10.1007/S10107-021-01744-W).
- [116] A. Natanzon, R. Shamir, R. Sharan, Complexity classification of some edge modification problems, *Discret. Appl. Math.* 113 (1) (2001) 109–128. doi:[10.1016/S0166-218X\(00\)00391-7](https://doi.org/10.1016/S0166-218X(00)00391-7).
- [117] R. Shamir, R. Sharan, D. Tsur, Cluster graph modification problems, *Discret. Appl. Math.* 144 (1-2) (2004) 173–182. doi:[10.1016/J.DAM.2004.01.007](https://doi.org/10.1016/J.DAM.2004.01.007).
- [118] J. Gramm, J. Guo, F. Hüffner, R. Niedermeier, Graph-modeled data clustering: Exact algorithms for clique generation, *Theory Comput. Syst.* 38 (4) (2005) 373–392. doi:[10.1007/S00224-004-1178-Y](https://doi.org/10.1007/S00224-004-1178-Y).
- [119] S. Böcker, P. Damaschke, Even faster parameterized cluster deletion and cluster editing, *Inf. Process. Lett.* 111 (14) (2011) 717–721. doi:[10.1016/J.IPL.2011.05.003](https://doi.org/10.1016/J.IPL.2011.05.003).
- [120] G. F. Italiano, A. L. Konstantinidis, C. Papadopoulos, Structural parameterization of cluster deletion, in: C. Lin, B. M. T. Lin, G. Liotta (Eds.), *WALCOM: Algorithms and Computation - 17th International Conference and Workshops, WALCOM 2023, Hsinchu, Taiwan, March 22-24, 2023, Proceedings*, Vol. 13973 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 371–383. doi:[10.1007/978-3-031-27051-2_31](https://doi.org/10.1007/978-3-031-27051-2_31).
- [121] A. Dessmark, J. Jansson, A. Lingas, E. Lundell, M. Persson, On the approximability of maximum and minimum edge clique partition problems, *Int. J. Found. Comput. Sci.* 18 (2) (2007) 217–226. doi:[10.1142/S0129054107004656](https://doi.org/10.1142/S0129054107004656).
- [122] N. J. Pullman, Clique coverings of graphs—a survey, in: *Combinatorial Mathematics X: Proceedings of the Conference held in Adelaide, Australia, August 23–27, 1982*, Springer, 2006, pp. 72–85. doi:[10.1007/BFb0071509](https://doi.org/10.1007/BFb0071509).
- [123] X. Li, R. Zhou, Y. Dai, L. Chen, C. Liu, Q. He, Y. Yang, Mining maximal clique summary with effective sampling, in: J. Wang, K. Shim, X. Wu (Eds.), *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8–11, 2019*, IEEE, 2019, pp. 1198–1203. doi:[10.1109/ICDM.2019.00147](https://doi.org/10.1109/ICDM.2019.00147).

BIBLIOGRAPHY

- [124] X. Li, R. Zhou, L. Chen, Y. Zhang, C. Liu, Q. He, Y. Yang, Finding a summary for all maximal cliques, in: 37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021, IEEE, 2021, pp. 1344–1355. doi:[10.1109/ICDE51399.2021.00120](https://doi.org/10.1109/ICDE51399.2021.00120).
- [125] J. Orlin, Contentment in graph theory: covering graphs with cliques, in: *Indagationes Mathematicae (Proceedings)*, Vol. 80, Elsevier, 1977, pp. 406–424. doi:[10.1016/1385-7258\(77\)90055-5](https://doi.org/10.1016/1385-7258(77)90055-5).
- [126] A. Conte, R. Grossi, A. Marino, Large-scale clique cover of real-world networks, *Inf. Comput.* 270 (2020). doi:[10.1016/J.IC.2019.104464](https://doi.org/10.1016/J.IC.2019.104464).
- [127] J. Wu, C. Li, L. Jiang, J. Zhou, M. Yin, Local search for diversified top- k clique search problem, *Comput. Oper. Res.* 116 (2020) 104867. doi:[10.1016/J.COR.2019.104867](https://doi.org/10.1016/J.COR.2019.104867).
- [128] F. Hao, Z. Pei, L. T. Yang, Diversified top- k maximal clique detection in social internet of things, *Future Gener. Comput. Syst.* 107 (2020) 408–417. doi:[10.1016/J.FUTURE.2020.02.023](https://doi.org/10.1016/J.FUTURE.2020.02.023).
- [129] J. Xue, J. Zheng, K. He, C. Li, Y. Liu, Diverteam: An effective evolutionary algorithm for diversified top- k (weight) clique search problems, in: U. Endriss, F. S. Melo, K. Bach, A. J. B. Diz, J. M. Alonso-Moral, S. Barro, F. Heintz (Eds.), *ECAI 2024 - 27th European Conference on Artificial Intelligence*, 19-24 October 2024, Santiago de Compostela, Spain, Vol. 392, IOS Press, 2024, pp. 4108–4115. doi:[10.3233/FAIA240981](https://doi.org/10.3233/FAIA240981).
- [130] K. Jansen, P. Scheffler, G. Woeginger, The disjoint cliques problem, *RAIRO-Operations Research* 31 (1) (1997) 45–66. doi:[10.1051/ro/1997310100451](https://doi.org/10.1051/ro/1997310100451).
- [131] B. P. W. Ames, S. A. Vavasis, Convex optimization for the planted k -disjoint-clique problem, *Math. Program.* 143 (1-2) (2014) 299–337. doi:[10.1007/S10107-013-0733-1](https://doi.org/10.1007/S10107-013-0733-1).
- [132] J. Wu, C. M. Li, Y. Zhou, M. Yin, X. Xu, D. Niu, HEA-D: A hybrid evolutionary algorithm for diversified top- k weight clique search problem, in: L. D. Raedt (Ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, ijcai.org, 2022, pp. 4821–4827. doi:[10.24963/IJCAI.2022/668](https://doi.org/10.24963/IJCAI.2022/668).

BIBLIOGRAPHY

- [133] J. Wu, M. Yin, A restart local search for solving diversified top-k weight clique search problem, *Mathematics* 9 (21) (2021) 2674. doi:[10.3390/math9212674](https://doi.org/10.3390/math9212674).
- [134] J. Zhou, C. Li, Y. Zhou, M. Li, L. Liang, J. Wang, Solving diversified top-k weight clique search problem, *Sci. China Inf. Sci.* 64 (5) (2021). doi:[10.1007/S11432-020-3069-4](https://doi.org/10.1007/S11432-020-3069-4).
- [135] A. P. Mukherjee, P. Xu, S. Tirthapura, Enumeration of maximal cliques from an uncertain graph, *IEEE Trans. Knowl. Data Eng.* 29 (3) (2017) 543–555. doi:[10.1109/TKDE.2016.2527643](https://doi.org/10.1109/TKDE.2016.2527643).
- [136] R. Li, Q. Dai, G. Wang, Z. Ming, L. Qin, J. X. Yu, Improved algorithms for maximal clique search in uncertain networks, in: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8–11, 2019, IEEE, 2019, pp. 1178–1189. doi:[10.1109/ICDE.2019.00108](https://doi.org/10.1109/ICDE.2019.00108).
- [137] Q. Dai, R. Li, M. Liao, H. Chen, G. Wang, Fast maximal clique enumeration on uncertain graphs: A pivot-based approach, in: Z. G. Ives, A. Bonifati, A. E. Abbadi (Eds.), *SIGMOD '22: International Conference on Management of Data*, Philadelphia, PA, USA, June 12 – 17, 2022, ACM, 2022, pp. 2034–2047. doi:[10.1145/3514221.3526143](https://doi.org/10.1145/3514221.3526143).
- [138] Z. Zou, J. Li, H. Gao, S. Zhang, Finding top-k maximal cliques in an uncertain graph, in: 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), 2010, pp. 649–652. doi:[10.1109/ICDE.2010.5447891](https://doi.org/10.1109/ICDE.2010.5447891).
- [139] J. Bai, J. Zhou, M. Du, Z. Chen, Index-based top $k\alpha$ -maximal-clique enumeration over uncertain graphs, *J. Supercomput.* 78 (17) (2022) 19372–19400. doi:[10.1007/S11227-022-04613-1](https://doi.org/10.1007/S11227-022-04613-1).
- [140] K. Rozman, A. Ghysels, D. Janežič, J. Konc, An exact algorithm to find a maximum weight clique in a weighted undirected graph, *Scientific reports* 14 (1) (2024) 9118. doi:[10.1038/s41598-024-59689-x](https://doi.org/10.1038/s41598-024-59689-x).
- [141] W. J. Pullan, Approximating the maximum vertex/edge weighted clique using local search, *J. Heuristics* 14 (2) (2008) 117–134. doi:[10.1007/S10732-007-9026-2](https://doi.org/10.1007/S10732-007-9026-2).
- [142] D. Yu, L. Zhang, Q. Luo, X. Cheng, Z. Cai, Maximal clique search in weighted graphs, *IEEE Trans. Knowl. Data Eng.* 35 (9) (2023) 9421–9432. doi:[10.1109/TKDE.2023.3239409](https://doi.org/10.1109/TKDE.2023.3239409).

BIBLIOGRAPHY

- [143] S. Shimizu, K. Yamaguchi, S. Masuda, A maximum edge-weight clique extraction algorithm based on branch-and-bound, *Discret. Optim.* 37 (2020) 100583. doi:[10.1016/J.DISOPT.2020.100583](https://doi.org/10.1016/J.DISOPT.2020.100583).
- [144] C. Lu, J. X. Yu, H. Wei, Y. Zhang, Enumerating maximum cliques in massive graphs, *IEEE Trans. Knowl. Data Eng.* 34 (9) (2022) 4215–4230. doi:[10.1109/TKDE.2020.3036013](https://doi.org/10.1109/TKDE.2020.3036013).
- [145] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
- [146] A. Buchanan, J. L. Walteros, S. Butenko, P. M. Pardalos, Solving maximum clique in sparse graphs: an $O(nm + n^{2^{d/4}})$ algorithm for d -degenerate graphs, *Optim. Lett.* 8 (5) (2014) 1611–1617. doi:[10.1007/S11590-013-0698-2](https://doi.org/10.1007/S11590-013-0698-2).
- [147] U. Feige, Approximating maximum clique by removing subgraphs, *SIAM J. Discret. Math.* 18 (2) (2004) 219–225. doi:[10.1137/S089548010240415X](https://doi.org/10.1137/S089548010240415X).
- [148] C. Komusiewicz, Multivariate algorithmics for finding cohesive subnetworks, *Algorithms* 9 (1) (2016) 21. doi:[10.3390/A9010021](https://doi.org/10.3390/A9010021).
- [149] B. Balasundaram, F. M. Pajouh, Graph theoretic clique relaxations and applications, *Handbook of combinatorial optimization* (2013) 1559–1598 doi:[10.1007/978-1-4419-7997-1_9](https://doi.org/10.1007/978-1-4419-7997-1_9).
- [150] P. M. Pardalos, J. Xue, The maximum clique problem, *Journal of global Optimization* 4 (1994) 301–328. doi:[10.1007/BF01098364](https://doi.org/10.1007/BF01098364).
- [151] I. M. Bomze, M. Budinich, P. M. Pardalos, M. Pelillo, The maximum clique problem, *Handbook of Combinatorial Optimization: Supplement Volume A* (1999) 1–74 doi:[10.1007/978-1-4757-3023-4_1](https://doi.org/10.1007/978-1-4757-3023-4_1).
- [152] Q. Wu, J. Hao, A review on algorithms for maximum clique problems, *Eur. J. Oper. Res.* 242 (3) (2015) 693–709. doi:[10.1016/J.EJOR.2014.09.064](https://doi.org/10.1016/J.EJOR.2014.09.064).
- [153] J. D. Eblen, C. A. Phillips, G. L. Rogers, M. A. Langston, The maximum clique enumeration problem: algorithms, applications, and implementations, *BMC Bioinform.* 13 (S-10) (2012) S5. doi:[10.1186/1471-2105-13-S10-S5](https://doi.org/10.1186/1471-2105-13-S10-S5).

BIBLIOGRAPHY

- [154] S. B. Seidman, Network structure and minimum degree, *Social networks* 5 (3) (1983) 269–287. doi:[10.1016/0378-8733\(83\)90028-X](https://doi.org/10.1016/0378-8733(83)90028-X).
- [155] V. Batagelj, M. Zaversnik, An $o(m)$ algorithm for cores decomposition of networks, *CoRR* cs.DS/0310049 (2003). doi:[10.48550/arXiv.cs/0310049](https://doi.org/10.48550/arXiv.cs/0310049).
- [156] X. Li, R. Zhou, L. Chen, C. Liu, Q. He, Y. Yang, Efficiently finding a maximal clique summary via effective sampling, *CoRR* abs/2009.10376 (2020). [arXiv:2009.10376](https://arxiv.org/abs/2009.10376).
- [157] S. Cai, K. Su, A. Sattar, Local search with edge weighting and configuration checking heuristics for minimum vertex cover, *Artif. Intell.* 175 (9-10) (2011) 1672–1696. doi:[10.1016/J.ARTINT.2011.03.003](https://doi.org/10.1016/J.ARTINT.2011.03.003).
- [158] S. Cai, Balance between complexity and quality: Local search for minimum vertex cover in massive graphs, in: Q. Yang, M. J. Wooldridge (Eds.), *Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, AAAI Press, 2015*, pp. 747–753.
- [159] J. Wu, C. Li, L. Jiang, J. Zhou, M. Yin, Personal communication (2024).
- [160] F. Hao, G. Min, Z. Pei, D. Park, L. T. Yang, K-clique community detection in social networks based on formal concept analysis, *IEEE Syst. J.* 11 (1) (2017) 250–259. doi:[10.1109/JSYST.2015.2433294](https://doi.org/10.1109/JSYST.2015.2433294).
- [161] H. Ito, K. Iwama, T. Osumi, Linear-time enumeration of isolated cliques, in: *European Symp. on Algorithms - ESA*, Vol. 3669 of LNCS, 2005, pp. 119–130. doi:[10.1007/11561071_13](https://doi.org/10.1007/11561071_13).
- [162] H. Ito, K. Iwama, Enumeration of isolated cliques and pseudo-cliques, *ACM Trans. Algorithms* 5 (4) (2009) 40:1–40:21. doi:[10.1145/1597036.1597044](https://doi.org/10.1145/1597036.1597044).
- [163] H. Molter, R. Niedermeier, M. Renken, Isolation concepts applied to temporal clique enumeration, *Netw. Sci.* 9 (S1) (2021) S83–S105. doi:[10.1017/NWS.2020.38](https://doi.org/10.1017/NWS.2020.38).
- [164] J. Gao, F. Hao, E. Yang, Y. Yang, G. Min, Concept stability based isolated maximal cliques detection in dynamic social networks, in: *Green, Pervasive, and Cloud Computing*, Vol. 12398 of LNCS, 2020, pp. 131–144. doi:[10.1007/978-3-030-64243-3_11](https://doi.org/10.1007/978-3-030-64243-3_11).

BIBLIOGRAPHY

- [165] Y. Okubo, M. Haraguchi, E. Tomita, Enumerating maximal isolated cliques based on vertex-dependent connection lower bound, in: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition - 12th International Conference, MLDM 2016, New York, NY, USA, July 16–21, 2016, Proceedings, Vol. 9729 of Lecture Notes in Computer Science, Springer, 2016, pp. 569–583. doi:[10.1007/978-3-319-41920-6_45](https://doi.org/10.1007/978-3-319-41920-6_45).
- [166] Y. Uno, Y. Ota, A. Uemichi, Web structure mining by isolated stars, in: Algorithms and Models for the Web-Graph, Vol. 4936 of LNCS, 2006, pp. 149–156. doi:[10.1007/978-3-540-78808-9_14](https://doi.org/10.1007/978-3-540-78808-9_14).
- [167] Z. Alamgir, S. Karim, S. Husnine, Linear-time algorithm for generating c-isolated bicliques, Int. J. Comput. Math. 94 (8) (2017) 1574–1590. doi:[10.1080/00207160.2016.1226498](https://doi.org/10.1080/00207160.2016.1226498).
- [168] F. L. Gall, O. Nadler, H. Nishimura, R. Oshman, Quantum simultaneous protocols without public coins using modified equality queries, in: 28th Int. Conf. on Principles of Distributed Systems, Vol. 324 of LIPIcs, 2024, pp. 34:1–34:20. doi:[10.4230/LIPICS.OPODIS.2024.34](https://doi.org/10.4230/LIPICS.OPODIS.2024.34).
- [169] M. D’Elia, I. Finocchi, M. Patrignani, Engineering algorithms for ℓ -isolated maximal clique enumeration, in: E. D. Giacomo, D. Mondal (Eds.), WALCOM: Algorithms and Computation - 20th International Conference and Workshops on Algorithms and Computation, WALCOM 2026, Perugia, Italy, March 4–6, 2026, Proceedings, Vol. 16444 of Lecture Notes in Computer Science, Springer, 2026, pp. 545–559. doi:[10.1007/978-981-95-7127-7_36](https://doi.org/10.1007/978-981-95-7127-7_36).
- [170] D. Avis, K. Fukuda, Reverse search for enumeration, Discrete Applied Mathematics 65 (1) (1996) 21–46, first International Colloquium on Graphs and Optimization. doi:[https://doi.org/10.1016/0166-218X\(95\)00026-N](https://doi.org/10.1016/0166-218X(95)00026-N).
- [171] A. Conte, R. Grossi, A. Marino, T. Uno, L. Versari, Proximity search for maximal subgraph enumeration, SIAM J. Comput. 51 (5) (2022) 1580–1625. doi:[10.1137/20M1375048](https://doi.org/10.1137/20M1375048).
- [172] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (2002) 47–97. doi:[10.1103/RevModPhys.74.47](https://doi.org/10.1103/RevModPhys.74.47).
- [173] M. Behrisch, A. Taraz, Efficiently covering complex networks with cliques of similar vertices, Theor. Comput. Sci. 355 (1) (2006) 37–47. doi:[10.1016/J.TCS.2005.12.005](https://doi.org/10.1016/J.TCS.2005.12.005).

BIBLIOGRAPHY

- [174] A. Conte, R. Grossi, A. Marino, T. Uno, L. Versari, Proximity search for maximal subgraph enumeration, *SIAM J. Comput.* 51 (5) (2022) 1580–1625. doi:[10.1137/20M1375048](https://doi.org/10.1137/20M1375048).
- [175] J. Wang, J. Cheng, Truss decomposition in massive networks, *Proc. VLDB Endow.* 5 (9) (2012) 812–823.
- [176] I. Finocchi, M. Finocchi, E. G. Fusco, Clique counting in mapreduce: Algorithms and experiments, *ACM J. Exp. Algorithmics* 20 (2015) 1.7:1–1.7:20. doi:[10.1145/2794080](https://doi.org/10.1145/2794080).
- [177] I. Finocchi, R. L. Garcia, B. Sinaireri, From stars to diamonds: Counting and listing almost complete subgraphs in large networks, *Comput. J.* 67 (6) (2024) 2151–2161. doi:[10.1093/COMJNL/BXAD129](https://doi.org/10.1093/COMJNL/BXAD129).
- [178] B. Srinivasan, B. Ribeiro, On the equivalence between positional node embeddings and structural graph representations, in: *ICLR 2020, Open-Review.net*, , 2020.
- [179] P. Angelini, W. S. Evans, F. Frati, J. Gudmundsson, SEFE without mapping via large induced outerplane graphs in plane graphs, *J. Graph Theory* 82 (1) (2016) 45–64. doi:[10.1002/JGT.21884](https://doi.org/10.1002/JGT.21884).
- [180] F. Dross, M. Montassier, A. Pinlou, Large induced forests in planar graphs with girth 4, *Discret. Appl. Math.* 254 (2019) 96–106. doi:[10.1016/J.DAM.2018.06.029](https://doi.org/10.1016/J.DAM.2018.06.029).
- [181] F. Dross, M. Montassier, A. Pinlou, A lower bound on the order of the largest induced linear forest in triangle-free planar graphs, *Discret. Math.* 342 (4) (2019) 943–950. doi:[10.1016/J.DISC.2018.11.023](https://doi.org/10.1016/J.DISC.2018.11.023).
- [182] R. Lukot’ka, J. Mazák, X. Zhu, Maximum 4-degenerate subgraph of a planar graph, *Electron. J. Comb.* 22 (1) (2015) 1. doi:[10.37236/4265](https://doi.org/10.37236/4265).
- [183] L. Kowalik, B. Luzar, R. Skrekovski, An improved bound on the largest induced forests for triangle-free planar graphs, *Discret. Math. Theor. Comput. Sci.* 12 (1) (2010) 87–100. doi:[10.46298/DMTCS.487](https://doi.org/10.46298/DMTCS.487).
- [184] H. Le, A better bound on the largest induced forests in triangle-free planar graph, *Graphs Comb.* 34 (6) (2018) 1217–1246. doi:[10.1007/S00373-018-1944-2](https://doi.org/10.1007/S00373-018-1944-2).

BIBLIOGRAPHY

- [185] M. R. Salavatipour, Large induced forests in triangle-free planar graphs, *Graphs Comb.* 22 (1) (2006) 113–126. doi:[10.1007/S00373-006-0642-7](https://doi.org/10.1007/S00373-006-0642-7).
- [186] L. Shi, H. Xu, Large induced forests in graphs, *J. Graph Theory* 85 (4) (2017) 759–779. doi:[10.1002/JGT.22104](https://doi.org/10.1002/JGT.22104).
- [187] Y. Wang, Q. Xie, X. Yu, Induced forests in bipartite planar graphs, *Journal of Combinatorics* 8 (1) (2017) 93–166.
- [188] K. Hosono, Induced forests in trees and outerplanar graphs, *Proceedings of the Faculty of Science of Tokai University* 25 (1990) 27–29.
- [189] O. V. Borodin, A proof of Grünbaum’s conjecture on the acyclic 5-colorability of planar graphs, in: *Doklady Akademii Nauk*, Vol. 231, Russian Academy of Sciences, 1976, pp. 18–20.
- [190] P. Bose, V. Dujmovic, H. Houdrouge, P. Morin, S. Odak, Connected dominating sets in triangulations, *CoRR* abs/2312.03399 (2023). doi:[10.48550/ARXIV.2312.03399](https://doi.org/10.48550/ARXIV.2312.03399).
- [191] M. O. Albertson, D. M. Berman, J. P. Hutchinson, C. Thomassen, Graphs with homeomorphically irreducible spanning trees, *J. Graph Theory* 14 (2) (1990) 247–258. doi:[10.1002/JGT.3190140212](https://doi.org/10.1002/JGT.3190140212).
- [192] K. Enami, N. Matsumoto, T. Yashima, Contributions to conjectures in planar graphs: Induced substructures, treewidth, and dominating sets, *CoRR* abs/2506.10471 (2025). doi:[10.48550/ARXIV.2506.10471](https://doi.org/10.48550/ARXIV.2506.10471).
- [193] K. S. Poh, On the linear vertex-arboricity of a planar graph, *J. Graph Theory* 14 (1) (1990) 73–75. doi:[10.1002/JGT.3190140108](https://doi.org/10.1002/JGT.3190140108).
- [194] N. Alon, D. Mubayi, R. Thomas, Large induced forests in sparse graphs, *J. Graph Theory* 38 (3) (2001) 113–123. doi:[10.1002/JGT.1028](https://doi.org/10.1002/JGT.1028).
- [195] G. Fertin, E. Godard, A. Raspaud, Minimum feedback vertex set and acyclic coloring, *Inf. Process. Lett.* 84 (3) (2002) 131–139. doi:[10.1016/S0020-0190\(02\)00265-X](https://doi.org/10.1016/S0020-0190(02)00265-X).
- [196] G. G. Chappell, M. J. Pelsmayer, Maximum induced forests in graphs of bounded treewidth, *Electron. J. Comb.* 20 (4) (2013) 8. doi:[10.37236/3826](https://doi.org/10.37236/3826).

BIBLIOGRAPHY

- [197] J. A. Bondy, G. Hopkins, W. Staton, Lower bounds for induced forests in cubic graphs, *Canad. Math. Bull.* 30 (2) (1987) 175–178.
- [198] T. Kelly, C. Liu, Size of the largest induced forest in subcubic graphs of girth at least four and five, *J. Graph Theory* 89 (4) (2018) 457–478. doi:[10.1002/JGT.22361](https://doi.org/10.1002/JGT.22361).
- [199] W. Staton, Induced forests in cubic graphs, *Discret. Math.* 49 (2) (1984) 175–178. doi:[10.1016/0012-365X\(84\)90115-8](https://doi.org/10.1016/0012-365X(84)90115-8).
- [200] Z. Dvoraák, T. Kelly, Induced 2-degenerate subgraphs of triangle-free planar graphs, *Electron. J. Comb.* 25 (1) (2018) 1. doi:[10.37236/7311](https://doi.org/10.37236/7311).
- [201] Y. Gu, H. A. Kierstead, S. Oum, H. Qi, X. Zhu, 3-degenerate induced subgraph of a planar graph, *J. Graph Theory* 99 (2) (2022) 251–277. doi:[10.1002/JGT.22740](https://doi.org/10.1002/JGT.22740).
- [202] R. J. Kang, M. Mnich, T. Müller, Induced matchings in subcubic planar graphs, *SIAM J. Discret. Math.* 26 (3) (2012) 1383–1411. doi:[10.1137/100808824](https://doi.org/10.1137/100808824).
- [203] M. D’Elia, F. Frati, Large induced subgraphs of bounded degree in outerplanar and planar graphs, in: E. D. Giacomo, D. Mondal (Eds.), *WALCOM: Algorithms and Computation - 20th International Conference and Workshops on Algorithms and Computation*, WALCOM 2026, Perugia, Italy, March 4–6, 2026, *Proceedings*, Vol. 16444 of *Lecture Notes in Computer Science*, Springer, 2026, pp. 187–202. doi:[10.1007/978-981-95-7127-7_13](https://doi.org/10.1007/978-981-95-7127-7_13).
- [204] N. Fountoulakis, R. J. Kang, C. McDiarmid, The t-stability number of a random graph, *Electron. J. Comb.* 17 (1) (2010). doi:[10.37236/331](https://doi.org/10.37236/331).
- [205] M. O. Albertson, A lower bound for the independence number of a planar graph, *J. Comb. Th. Ser. B* 20 (1) (1976) 84–93. doi:[https://doi.org/10.1016/0095-8956\(76\)90071-X](https://doi.org/10.1016/0095-8956(76)90071-X).
- [206] K. Appel, W. Haken, Every planar map is four colorable. I. Discharging, *Illinois J. Math.* 21 (3) (1977) 429–490. doi:[doi:10.1215/ijm/1256049011](https://doi.org/10.1215/ijm/1256049011).
- [207] K. Appel, W. Haken, J. Koch, Every planar map is four colorable. II. Reducibility, *Illinois J. Math.* 21 (3) (1977) 491–567. doi:[doi:10.1215/ijm/1256049012](https://doi.org/10.1215/ijm/1256049012).

BIBLIOGRAPHY

- [208] T. Biedl, D. F. Wilkinson, Bounded-degree independent sets in planar graphs, *Theory Comput. Syst.* 38 (3) (2005) 253–278. doi:[10.1007/S00224-005-1139-0](https://doi.org/10.1007/S00224-005-1139-0).
- [209] Y. Caro, Y. Roditty, On the vertex-independence number and star decomposition of graphs, *Ars Combin.* 20 (1985) 167–180.
- [210] N. Robertson, D. P. Sanders, P. D. Seymour, R. Thomas, The four-colour theorem, *J. Comb. Theory B* 70 (1) (1997) 2–44. doi:[10.1006/JCTB.1997.1750](https://doi.org/10.1006/JCTB.1997.1750).
- [211] L. J. Cowen, R. Cowen, D. R. Woodall, Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency, *J. Graph Theory* 10 (2) (1986) 187–195. doi:[10.1002/JGT.3190100207](https://doi.org/10.1002/JGT.3190100207).
- [212] D. R. Wood, Defective and clustered graph colouring, *Electron. J. Comb.* #DS23 (2018) 1–71. doi:[10.37236/7406](https://doi.org/10.37236/7406).
- [213] D. G. Kirkpatrick, Optimal search in planar subdivisions, *SIAM J. Comput.* 12 (1) (1983) 28–35. doi:[10.1137/0212002](https://doi.org/10.1137/0212002).
- [214] J. Snoeyink, M. J. van Kreveld, Linear-time reconstruction of Delaunay triangulations with applications, in: R. E. Burkard, G. J. Woeginger (Eds.), 5th Annual European Symposium on Algorithms (ESA '97), Vol. 1284 of LNCS, Springer, 1997, pp. 459–471. doi:[10.1007/3-540-63397-9_35](https://doi.org/10.1007/3-540-63397-9_35).
- [215] P. Bose, V. Dujmovic, D. R. Wood, Induced subgraphs of bounded degree and bounded treewidth, *Contributions Discret. Math.* 1 (1) (2006). doi:[10.11575/CDM.V1I1.61915](https://doi.org/10.11575/CDM.V1I1.61915).
- [216] K. Knauer, T. Ueckerdt, Clustered independence and bounded treewidth, *CoRR* abs/2303.13655 (2023). doi:[10.48550/ARXIV.2303.13655](https://doi.org/10.48550/ARXIV.2303.13655).
- [217] H. Liu, L. Sun, On domination number of 4-regular graphs, *Czechoslovak Mathematical Journal* 54 (2004) 889–898. doi:[10.1007/s10587-004-6438-0](https://doi.org/10.1007/s10587-004-6438-0).
- [218] H.-M. Xing, L. Sun, X.-G. Chen, An upper bound for domination number of 5-regular graphs, *Czechoslovak Mathematical Journal* 56 (2006) 1049–1061. doi:[10.1007/s10587-006-0079-4](https://doi.org/10.1007/s10587-006-0079-4).

BIBLIOGRAPHY

- [219] M. D’Elia, F. Frati, On large induced outerplanar subgraphs in 2-outerplanar graphs, CoRR abs/2602.18121 (2026). doi:[10.48550/ARXIV.2602.18121](https://doi.org/10.48550/ARXIV.2602.18121).
- [220] S. Arnborg, A. Proskurowski, Linear time algorithms for np-hard problems restricted to partial k-trees, Discret. Appl. Math. 23 (1) (1989) 11–24. doi:[10.1016/0166-218X\(89\)90031-0](https://doi.org/10.1016/0166-218X(89)90031-0).
- [221] D. G. Corneil, U. Rotics, On the relationship between clique-width and treewidth, SIAM J. Comput. 34 (4) (2005) 825–847. doi:[10.1137/S0097539701385351](https://doi.org/10.1137/S0097539701385351).
- [222] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015. doi:[10.1007/978-3-319-21275-3](https://doi.org/10.1007/978-3-319-21275-3).
- [223] N. Alon, J. H. Spencer, The Probabilistic Method, Third Edition, Wiley-Interscience series in discrete mathematics and optimization, Wiley, 2008.
- [224] N. Castañeda, J. Urrutia, [Straight line embeddings of planar graphs on point sets](#), in: F. Fiala, E. Kranakis, J. Sack (Eds.), 8th Canadian Conference on Computational Geometry (CCCG 1996), Carleton University Press, 1996, pp. 312–318.
URL http://www.cccg.ca/proceedings/1996/cccg1996_0052.pdf
- [225] P. Gritzmann, B. Mohar, J. Pach, R. Pollack, Embedding a planar triangulation with vertices at specified points, The American Mathematical Monthly 98 (2) (1991) 165. doi:[10.2307/2323956](https://doi.org/10.2307/2323956).
- [226] P. Bose, On embedding an outer-planar graph in a point set, in: G. Di Battista (Ed.), 5th International Symposium on Graph Drawing (GD ’97), Vol. 1353 of LNCS, Springer, 1997, pp. 25–36. doi:[10.1007/3-540-63938-1_47](https://doi.org/10.1007/3-540-63938-1_47).
- [227] P. Bose, On embedding an outer-planar graph in a point set, Comput. Geom. 23 (3) (2002) 303–312. doi:[10.1016/S0925-7721\(01\)00069-4](https://doi.org/10.1016/S0925-7721(01)00069-4).
- [228] P. Bose, M. McAllister, J. Snoeyink, Optimal algorithms to embed trees in a point set, in: F. Brandenburg (Ed.), Symposium on Graph Drawing (GD ’95), Vol. 1027 of LNCS, Springer, 1995, pp. 64–75. doi:[10.1007/BFB0021791](https://doi.org/10.1007/BFB0021791).

BIBLIOGRAPHY

- [229] P. Bose, M. McAllister, J. Snoeyink, Optimal algorithms to embed trees in a point set, *J. Graph Algorithms Appl.* 1 (2) (1997) 1–15. doi:[10.7155/JGAA.00002](https://doi.org/10.7155/JGAA.00002).
- [230] S. Cabello, Planar embeddability of the vertices of a graph using a fixed point set is NP-hard, *J. Graph Algorithms Appl.* 10 (2) (2006) 353–363. doi:[10.7155/JGAA.00132](https://doi.org/10.7155/JGAA.00132).
- [231] R. I. Nishat, D. Mondal, M. S. Rahman, Point-set embeddings of plane 3-trees - (extended abstract), in: U. Brandes, S. Cornelsen (Eds.), 18th International Symposium on Graph Drawing (GD 2010), Vol. 6502 of LNCS, Springer, 2010, pp. 317–328. doi:[10.1007/978-3-642-18469-7_29](https://doi.org/10.1007/978-3-642-18469-7_29).
- [232] R. I. Nishat, D. Mondal, M. S. Rahman, Point-set embeddings of plane 3-trees, *Comput. Geom.* 45 (3) (2012) 88–98. doi:[10.1016/J.COMGEO.2011.09.002](https://doi.org/10.1016/J.COMGEO.2011.09.002).
- [233] T. Biedl, M. Vatschelle, [The point-set embeddability problem for plane graphs](#), in: T. K. Dey, S. Whitesides (Eds.), 28th ACM Symposium on Computational Geometry (SoCG 2012), ACM, 2012, pp. 41–50. doi:[10.1145/2261250.2261257](https://doi.org/10.1145/2261250.2261257). URL <https://doi.org/10.1145/2261250.2261257>
- [234] G. Di Battista, W. Didimo, L. Grilli, F. Grosso, G. Ortali, M. Patrignani, A. Tappini, Small point-sets supporting graph stories, in: P. Angelini, R. von Hanxleden (Eds.), 30th International Symposium on Graph Drawing and Network Visualization (GD 2022), Vol. 13764 of LNCS, Springer, 2022, pp. 289–303. doi:[10.1007/978-3-031-22203-0_21](https://doi.org/10.1007/978-3-031-22203-0_21).
- [235] C. Binucci, E. Di Giacomo, W. Didimo, A. Estrella-Balderrama, F. Frati, S. G. Kobourov, G. Liotta, Upward straight-line embeddings of directed graphs into point sets, *Comput. Geom.* 43 (2) (2010) 219–232. doi:[10.1016/J.COMGEO.2009.07.002](https://doi.org/10.1016/J.COMGEO.2009.07.002).
- [236] L. S. Heath, S. V. Pemmaraju, A. N. Trenk, Stack and queue layouts of directed acyclic graphs: Part I, *SIAM J. Comput.* 28 (4) (1999) 1510–1539. doi:[10.1137/S0097539795280287](https://doi.org/10.1137/S0097539795280287).
- [237] P. Angelini, F. Frati, M. Geyer, M. Kaufmann, T. Mchedlidze, A. Symvonis, Upward geometric graph embeddings into point sets, in: U. Brandes, S. Cornelsen (Eds.), 18th International Symposium on Graph

BIBLIOGRAPHY

- Drawing (GD 2010), Vol. 6502 of LNCS, Springer, 2010, pp. 25–37. doi:10.1007/978-3-642-18469-7_3.
- [238] T. Mchedlidze, Upward planar embedding of an n -vertex oriented path on $O(n^2)$ points, *Comput. Geom.* 46 (8) (2013) 1003–1008. doi:10.1016/J.COMGEO.2013.05.004.
- [239] E. Di Giacomo, H. Förster, D. Kokhovich, T. Mchedlidze, F. Montecchiani, A. Symvonis, A. Villedieu, On 1-bend upward point-set embeddings of st-digraphs, in: J. A. Soto, A. Wiese (Eds.), 16th Latin American Symposium on Theoretical Informatics (LATIN 2024), Part I, Vol. 14578 of LNCS, Springer, 2024, pp. 3–18. doi:10.1007/978-3-031-55598-5_1.
- [240] F. Giordano, G. Liotta, T. Mchedlidze, A. Symvonis, S. Whitesides, Computing upward topological book embeddings of upward planar digraphs, *J. Discrete Algorithms* 30 (2015) 45–69. doi:10.1016/J.JDA.2014.11.006.
- [241] M. Kaufmann, R. Wiese, Embedding vertices at points: Few bends suffice for planar graphs, in: J. Kratochvíl (Ed.), 7th International Symposium on Graph Drawing (GD’99), Vol. 1731 of LNCS, Springer, 1999, pp. 165–174. doi:10.1007/3-540-46648-7_17.
- [242] M. Kaufmann, R. Wiese, Embedding vertices at points: Few bends suffice for planar graphs, *J. Graph Algorithms Appl.* 6 (1) (2002) 115–129. doi:10.7155/JGAA.00046.
- [243] M. Kaufmann, T. Mchedlidze, A. Symvonis, Upward point set embeddability for convex point sets is in P, in: M. J. van Kreveld, B. Speckmann (Eds.), 19th International Symposium on Graph Drawing (GD 2011), Vol. 7034 of LNCS, Springer, 2011, pp. 403–414. doi:10.1007/978-3-642-25878-7_38.
- [244] M. Kaufmann, T. Mchedlidze, A. Symvonis, On upward point set embeddability, *Comput. Geom.* 46 (6) (2013) 774–804. doi:10.1016/J.COMGEO.2012.11.008.
- [245] C. Alegría, S. Caroppo, G. Da Lozzo, M. D’Elia, G. Di Battista, F. Frati, F. Grosso, M. Patrignani, Upward pointset embeddings of planar st-graphs, in: S. Felsner, K. Klein (Eds.), 32nd International Symposium on Graph Drawing and Network Visualization (GD 2024), Vol. 320 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 24:1–24:18. doi:10.4230/LIPICS.GD.2024.24.

BIBLIOGRAPHY

- [246] C. Alegría-Galicia, S. Caroppo, G. D. Lozzo, M. D’Elia, G. D. Battista, F. Frati, F. Grosso, M. Patrignani, Upward pointset embeddings of planar st-graphs, *Algorithmica* 87 (6) (2025) 930–960. doi:[10.1007/S00453-025-01302-2](https://doi.org/10.1007/S00453-025-01302-2).
- [247] E. Arseneva, P. Cano, L. Kleist, T. Mchedlidze, S. Mehrabi, I. Parada, P. Valtr, Upward point set embeddings of paths and trees, in: R. Uehara, S. Hong, S. C. Nandy (Eds.), 15th International Conference and Workshops on Algorithms and Computation (WALCOM 2021), Vol. 12635 of LNCS, Springer, 2021, pp. 234–246. doi:[10.1007/978-3-030-68211-8_19](https://doi.org/10.1007/978-3-030-68211-8_19).
- [248] G. Di Battista, R. Tamassia, Algorithms for plane representations of acyclic digraphs, *Theor. Comput. Sci.* 61 (1988) 175–198. doi:[10.1016/0304-3975\(88\)90123-5](https://doi.org/10.1016/0304-3975(88)90123-5).
- [249] V. Alvarez, K. Bringmann, R. Curticapean, S. Ray, Counting triangulations and other crossing-free structures via onion layers, *Discret. Comput. Geom.* 53 (4) (2015) 675–690. doi:[10.1007/S00454-015-9672-3](https://doi.org/10.1007/S00454-015-9672-3).
- [250] G. Cheon, H. J. Choi, G. Esteban, M. Song, Enumeration of bipartite non-crossing geometric graphs, *Discret. Appl. Math.* 317 (2022) 86–100. doi:[10.1016/J.DAM.2022.04.008](https://doi.org/10.1016/J.DAM.2022.04.008).
- [251] P. Flajolet, M. Noy, Analytic combinatorics of non-crossing configurations, *Discret. Math.* 204 (1-3) (1999) 203–229. doi:[10.1016/S0012-365X\(98\)00372-0](https://doi.org/10.1016/S0012-365X(98)00372-0).
- [252] D. Marx, T. Miltzow, Peeling and nibbling the cactus: Subexponential-time algorithms for counting triangulations and related problems, in: S. P. Fekete, A. Lubiw (Eds.), 32nd International Symposium on Computational Geometry (SoCG 2016), Vol. 51 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 52:1–52:16. doi:[10.4230/LIPICS.SOCG.2016.52](https://doi.org/10.4230/LIPICS.SOCG.2016.52).
- [253] A. Razen, E. Welzl, Counting plane graphs with exponential speed-up, in: C. S. Calude, G. Rozenberg, A. Salomaa (Eds.), *Rainbow of Computer Science - Dedicated to Hermann Maurer on the Occasion of His 70th Birthday*, Vol. 6570 of LNCS, Springer, 2011, pp. 36–46. doi:[10.1007/978-3-642-19391-0_3](https://doi.org/10.1007/978-3-642-19391-0_3).

BIBLIOGRAPHY

- [254] D. Eppstein, Counting polygon triangulations is hard, *Discret. Comput. Geom.* 64 (4) (2020) 1210–1234. doi:[10.1007/S00454-020-00251-7](https://doi.org/10.1007/S00454-020-00251-7).
- [255] J. S. B. Mitchell, J. O’Rourke, Computational geometry column 42, *Int. J. Comput. Geom. Appl.* 11 (5) (2001) 573–582. doi:[10.1142/S0218195901000651](https://doi.org/10.1142/S0218195901000651).
- [256] M. Wettstein, Counting and enumerating crossing-free geometric graphs, *J. Comput. Geom.* 8 (1) (2017) 47–77. doi:[10.20382/JOCG.V8I1A4](https://doi.org/10.20382/JOCG.V8I1A4).
- [257] K. Yamanaka, D. Avis, T. Horiyama, Y. Okamoto, R. Uehara, T. Yamauchi, Algorithmic enumeration of surrounding polygons, *Discret. Appl. Math.* 303 (2021) 305–313. doi:[10.1016/J.DAM.2020.03.034](https://doi.org/10.1016/J.DAM.2020.03.034).
- [258] D. Eppstein, Non-crossing hamiltonian paths and cycles in output-polynomial time, *Algorithmica* 86 (9) (2024) 3027–3053. doi:[10.1007/S00453-024-01255-Y](https://doi.org/10.1007/S00453-024-01255-Y).
- [259] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, first edition Edition, W. H. Freeman, 1979.
- [260] M. I. Shamos, D. Hoey, Geometric intersection problems, in: 17th Annual Symposium on Foundations of Computer Science (FOCS 1976), IEEE Computer Society, 1976, pp. 208–215. doi:[10.1109/SFCS.1976.16](https://doi.org/10.1109/SFCS.1976.16).
- [261] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 4th Edition, MIT Press, 2022.