Combinatorial algorithms for graphs and partially ordered sets

Johan Nilsson

PhD Dissertation

Department of Computer Science
University of Aarhus
Denmark

Combinatorial algorithms for graphs and partially ordered sets

A Dissertation
Presented to the Faculty of Science
of the University of Aarhus
in Partial Fulfillment of the Requirements for the
PhD Degree

by
Johan Nilsson
August 2, 2007

Abstract

In this thesis we study combinatorial and algorithmic problems for graphs and partially ordered sets. The thesis is centered around four themes: planar graphs, reachability, order dimension and special cases of NP-hard problems. The results presented are of both algorithmic and structural flavor. The thesis begins with an overview of the problems considered and an introduction to the necessary mathematical tools, whereafter the results are described in more detail.

Thorup proved that we can label the vertices in a planar directed n-vertex graph with labels of size $O(\log n)$, such that reachability queries can be answered in constant time by inspecting only the labels of the two vertices in the query. In Chapter 2 we generalize Thorup's labeling scheme to a setting where only a subset of the vertices are labeled. We prove that if the set of vertices we are interested in is a subset of the union of f faces, we can label the interesting vertices with labels of size $O(\log f)$, while still supporting constant time reachability queries. In particular, this yields an optimal labeling scheme for reachability in k-outerplanar digraphs for constant k.

In Chapter 3 we study the combinatorial problem of replacing a directed graph and a set of interesting vertices with a graph of smaller size while preserving the existence of paths between the interesting vertices. We call such a new graph a reachability substitute, and we prove that finding a reachability substitute of minimum size is NP-hard. Furthermore, we show that for almost all graphs and sets of interesting vertices, no planar reachability substitute exists, not of any size.

The dimension of a partially ordered set $\mathbf{P}$ is the minimum number of linear orders whose intersection is $\mathbf{P}$, and the vertex-edge-face poset $\mathbf{P}_M$ of a planar map M is the poset on the vertices, edges and faces of M ordered by inclusion. Brightwell and Trotter proved that $\dim(\mathbf{P}_M) \leq 4$. In Chapter 4 we investigate the cases where $\dim(\mathbf{P}_M) \leq 3$ and where $\dim(\mathbf{Q}_M) \leq 3$; here $\mathbf{Q}_M$ denotes the vertex-face poset of M, which is the subposet of $\mathbf{P}_M$ induced by the vertices and faces of M.

We show that a map M with $\dim(\mathbf{P}_M) \leq 3$ must be outerplanar and have an outerplanar dual. We concentrate on the simplest class of such maps and prove that within this class $\dim(\mathbf{P}_M) \leq 3$ is equivalent to the existence of a certain oriented coloring of a subset of the edges. This condition is easily checked and can be turned into a linear time algorithm returning a 3-realizer.

Additionally, we prove that if M is 2-connected and M and M^* are outerplanar, then $\dim(\mathbf{Q}_M) \leq 3$. There are, however, outerplanar maps with $\dim(\mathbf{Q}_M) = 4$. We construct the first such example.

Finally, in Chapter 5 we consider some classic NP-hard graph optimization problems, e.g., maximum independent set. Using a simple partitioning lemma, we extend algorithms that give exact solutions to these problem for graphs with small treewidth, to approximation algorithms for graphs with larger treewidth.

Acknowledgements

I would like to thank everyone who has helped and supported me during my PhD studies. First of all, I am grateful to my advisor Gerth Stølting Brodal. Although our interests didn't always match, he always provided good advice and encouraged me. One could not ask for a better advisor.

A special thanks goes to Stefan Felsner, who invited me to Technische Universität Berlin, and with whom I thoroughly enjoyed collaborating. I would also like to thank Irit Katriel, who introduced me to reachability problems, and all my other coauthors: Artur Czumaj, Magnus Halldórsson, Andrzej Lingas, Martin Kutz and Martin Skutella.

Thanks to the people in the Discrete Mathematics group at TU Berlin, who made my stay there very enjoyable: Florian, Conny, Sarah and Mareike. I have been lucky to share offices with some great people in Århus: Chris, Rune, Mirka, Troels, Claudio and Gabi. A big thank you also goes to all of the other BRICS PhD students who have helped making the last three years a fun experience, as well as to the administrative and technical staff at the Department of Computer Science who has solved every practical problem swiftly.

Finally, I acknowledge generous financial support from the BRICS research school and the Marie Curie Research Training Network COMBSTRU.

Johan Nilsson,
Århus, August 2, 2007.

Contents

List of figures

Chapter 1

Introduction

Many interesting and important problems lie in the intersection of graph theory and the theory of finite partially ordered sets. Studying those problems from a structural perspective leads to both new combinatorial and algorithmic insights.

Perhaps the most fundamental algorithmic graph problem is the reachability problem: given two vertices u and v in a graph G, is there a path from the u to v in G? Two problems related to reachability in directed graphs are considered. If we want to answer many reachability queries for the same graph, it makes sense to pre-compute the answers in order to improve the speed with which the queries can be answered. The result of such a pre-computation is called a reachability oracle.Of course, we could just store the answers to all possible queries in an $n \times n$ matrix, where n is the number of vertices in the graph, but ideally, we want a more compact data structure that still can provide quick answers to queries. In general, this is unfortunately not possible, but we will see that for planar graphs, we can do much better than storing the adjacency matrix.

Every acyclic digraph induces a poset on its vertices. There is a path from the vertex u to the vertex v in the graph if and only if $u < v$ in the corresponding poset. Hence, a reachability oracle for an acyclic graph is also a representation of some poset. It turns out that cycles are very easy to handle, so we can view the reachability oracle problem as the problem of efficiently representing posets.

If we have a graph were only some of the vertices are interesting, it would be nice if we could replace it with a smaller graph that preserves the existence of paths between interesting vertices. Here too we only have to consider acyclic graphs. We will see that this problem can also be formulated as the problem of finding a superposet with a cover graph of small size.

Every finite partially ordered set $\mathbf{P}$ is the intersection of finitely many linear orders. The minimum number of linear orders on the same ground set whose intersection is $\mathbf{P}$ is called the dimension of $\mathbf{P}$. Dimension has, since its introduction in 1941 [24], become a significant topic of research in combinatorics. There are many connections and analogies between dimension and the chromatic number of graphs and hypergraphs. Whereas the chromatic number in some sense measures how close to being a independent set a graph is, the dimension of a poset measures how close to being a linear order the poset is.

The posets induced by the incidence structures of graphs have attracted a lot of attention in the last two decades. Schnyder proved that the dimension of the poset induced by the incidences of vertices and edges of a graph has dimension at most 3 if and only if the graph is planar [53]. For planar maps, the dimension of the posets induced by the incidences of vertices and faces and vertices, edges and faces have also been studied. This is the problem considered in this thesis.

There is an interesting analogy between the chromatic number of planar graphs and the dimension of the vertex-face poset of planar maps. Every planar graph has chromatic number at most four, and every planar map has vertex-face dimension at most 4. Planar graphs with chromatic number 2 and planar maps with vertex-face dimension 2 has relatively simple structures. On the other hand, the maps with dimension 3 and the 3-colorable graphs do not seem to have nice characterizations. The latter case is NP-complete to recognize, while the complexity of the former case is still unknown. It is connected to a major open problem in dimension theory.

Finally, we study some special cases of NP-hard graph problems. Treewidth is a central concept in modern graph theory. It is also a very successful parameterization of graphs — many intractable problems become tractable in graphs with bounded treewidth. We introduce a scheme to produce approximation algorithms for graphs with larger treewidth.

1.1 Definitions

Before we proceed with the discussion of the contents of the thesis, we recall some definitions about graphs and partially ordered sets, and introduce some tools from dimension theory.

1.1.1 Graphs

We start by stating definitions of some basic graph theoretic terms relevant to this thesis. For a lengthier introduction, we refer to e.g. Diestel's excellent textbook [21].

An undirected graph $G = (V, E)$ is a pair of a set of vertices V and set of unordered pairs E of vertices. The vertices and edges of the graph G are also denoted $V(G)$ and $E(G)$, respectively. An undirected edge containing the vertices u and v is denoted $\{u, v\}$, and u and v are the *endpoints* of $\{u, v\}$. The vertices u and v are *adjacent*, and the edge $\{u, v\}$ is *incident* on u and v. The number of edges incident on a vertex v is called the *degree* of v. A *multigraph* is a graph where we allow multiple edges between the same endpoints, and edges with the two endpoints being identical (loops).

Directed graphs, also called *digraphs*, are pairs of sets of vertices and sets of ordered pairs of vertices. A directed edge (u, v) is an edge from the vertex u to the vertex v. We say that (u, v) has the *head* v and the *tail* u. The edge (u, v) is outgoing from u and incoming into v. The number of incoming edges into a vertex v is called the *indegree* of v and the number of outgoing edges is called the *outdegree* of v. Vertices with outdegree 0 are called *sinks* and a vertices with indegree 0 are called *sources*.

A graph $H = (V', E')$ is a *subgraph* of the graph G if $V' \subseteq V(G)$ and $E' \subseteq E(G)$. If G' contains all the edges of G between the vertices in V', G' is said to be an *induced subgraph* of G. The subgraph of G induced by the vertex set V' is written $G[V']$.

A *bipartite* graph is a graph $G = (A \cup B, E)$, such that all edges in E have one endpoint in A and one in B. The vertex sets A and B are called bipartitions. We call a graph with the maximum number of edges is a *complete graph*, and is denoted K_n if it has n vertices. A complete graph is also called a *clique*. A *complete bipartite graph* is a bipartite graph with the maximum number of edges. The complete bipartite graph with m vertices in one bipartition and n vertices in the other is denoted $K_{m,n}$.

Paths and cycles

A *path* is a graph of the form $V = \{v_1, v_2, v_3, \ldots, v_k\}$, $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}\}$. If $k \geq 3$ and we add the edge $\{v_k, v_1\}$, we get a *cycle*. The *length* of a path is the number of edges in it. The *distance* $d_G(u, v)$ from the vertex u to the vertex v in the graph G, is the length of the shortest path from u to v in G. A graph with no cycles in it is called *acyclic*.

Minors and subdivisions

We *contract* an edge $e = \{u, v\}$ in a graph G by removing e and identifying the vertices u and v. Contractions of directed edges are defined in the same way. A graph H is a *minor* of G if H can be obtained by contracting some of the edges of a subgraph of G.

A *subdivision* of an edge e is path of length at length at least 2 between the endpoints of e replacing it. A subdivision of a graph G is a graph obtained from G by subdividing one or more of the edges in $E(G)$. See Figure 1.1 for an example.

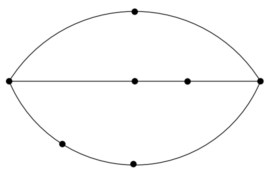

Figure 1.1: A subdivision of $K_{2,3}$.

Connectivity

An undirected graph G is *connected* if there is a path from each vertex to every other vertex, and it is *k-connected* if it still connected after any $k - 1$ vertices has been removed. A graph that is not connected is *disconnected*. Note that 1-connected just means connected. The maximal k-connected subgraphs of G are called the *k-connected components*. Sometimes, the connected components of G are just called the *components* of G.

If G is 1-connected or 2-connected, some vertices are of special interest. A *cutvertex* in a graph is a vertex whose removal makes the graph disconnected, while a *separating pair* is a pair of vertices who does the same. An edge whose removal disconnects the graph is called a *bridge*.

For directed graphs we have additional terms for connectivity. We say that a directed graph G is *strongly connected* if there is a directed path from each vertex $v \in V(G)$ to all the vertices in $V(G) \setminus \{v\}$, and it is *weakly connected* if the underlying undirected graph is connected.

Trees

An undirected acyclic graph is called a *tree*. If a specific vertex is designated *root* of a tree, the tree is said to be *rooted.* In this case, non-root vertices of degree 1 are called *leaves*. A *root path* is a path that ends in the root.

Planarity and planar maps

A *plane drawing* D of a graph G is a representation of G by points and arcs in the Euclidean plane in which two arcs meet only at common vertices. We say that a graph is *planar* if it has a plane drawing.

The regions of the plane that we get when we remove a plane drawing of G are called the faces of G. There is exactly one face which is unbounded, this is called the *outer face* If there is a set of faces $\mathcal{F}$ and a set of vertices U in a plane drawing of G such that $U \subseteq \cup\mathcal{F}$, we say that $\mathcal{F}$ *covers* U.

A graph G is *outerplanar* if it has a plane drawing where all vertices are on the unbounded face. Alternatively, we say that G is 1-outerplanar. We inductively define k-outerplanarity: G is k-outerplanar if it has a drawing where removing all the vertices on the unbounded face results in a $k-1$-outerplanar graph.

A *planar map* $M = (G, D)$ consists of a finite planar multigraph G and a plane drawing D of G. In this thesis a planar map M we will be understood as the combinatorial data given by the set V of vertices, the set E of edges, the set F of faces of M and the incidence relations between these sets. The *dual map* M^* of M is defined as follows: there is a vertex F^* in M^* for each face F in M, and an edge e^* in M^* for each edge e of M, joining the dual vertices corresponding to the faces in M separated by e (if e is a bridge, e^* is a loop). Each vertex in M will then correspond to a face of M^*.

Many of the maps we consider in the thesis are outerplanar. We differentiate between two notions of outerplanar maps. A planar map $M = (G, D)$ is *weakly outerplanar* if G is outerplanar, and *strongly outerplanar* if G is outerplanar and D is an outerplane drawing of G, i.e., a plane drawing of G where all the vertices are on the boundary of the outer face (see Figure 1.2). When it is clear from the context, the qualifiers weakly and strongly will be omitted.

1.1.2 Partially Ordered Sets

Finite partially ordered sets are basic combinatoric objects. Following Trotter [58], we introduce some basic terms in the partially ordered set literature.

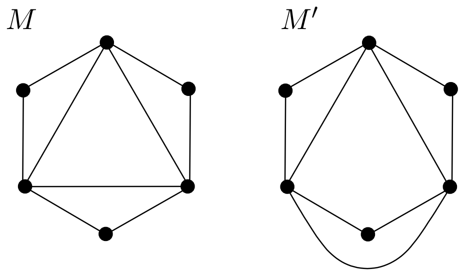

Figure 1.2: The map M is strongly outerplanar, while the map M' is weakly outerplanar.

A partially ordered set (poset) $\mathbf{P}$ is a pair (X, P), where X is a set (the *ground set*) and P is a reflexive, antisymmetric, and transitive binary relation on X. The relation P is then a partial order on X. We often do not distinguish between posets and partial orders when the ground set is clear from the context.

Throughout the thesis, the notations $x \leq y$ in P, $y \geq x$ in P and $(x, y) \in P$ are used interchangeably. We say $x < y$ in P and $y > x$ in P when $x \leq y$ in P and $x \neq y$. When $x, y \in X$, $(x, y) \notin P$ and $(y, x) \notin P$, x and y are *incomparable*. Furthermore, if $x < y$ in P, and there is no $z \in X \setminus \{x, y\}$ such that $x < z < y$ in P, y is said to *cover* x in P.

When Y is a subset of the ground set X, the restriction $P(Y)$ of the partial order P on X to Y is a partial order on Y. The poset $(Y, P(Y))$ is then called a *subposet* of (X, P), and (X, P) is called a *superposet* of $(Y, P(Y))$. The order $P^d = \{(y, x) \mid (x, y) \in P\}$ is called the *dual* of the partial order P.

A poset is called a *chain* if all elements are comparable, and an *antichain* if no two elements are comparable. If (X, P) is a chain, then P is called a *linear order*. The maximum cardinality of a chain in a poset $\mathbf{P}$ is called the *height* of $\mathbf{P}$.

The *cover graph* of a poset (X, P) is a graph $G = (X, E)$, where $\{x, y\} \in E$ if and only if y covers x or x covers y in P. A poset is often represented by an upward, straight line drawing the Euclidean plane of the directed graph obtained from the cover graph by orienting each edge towards the endpoint that covers the other endpoint. Such a drawing is called a *Hasse diagram*, or just a diagram.

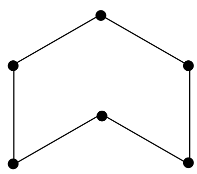

Figure 1.3: A Hasse diagram of a 3-dimensional poset of height 3.

A linear order L on X is called a *linear extension* of the partial order P on X when $x < y$ in L for all $x, y \in X$ with $x < y$ in P. A family $\mathcal{R}$ of linear extensions of P is called a *realizer* of P when $P = \bigcap \mathcal{R}$, i.e., for all $x, y \in X$, $x < y$ in P if and only if $x < y$ in every $L \in \mathcal{R}$. The *dimension* of $\mathbf{P}$, denoted dim($\mathbf{P}$), is the minimum cardinality of a realizer of P.

After seeing the definition of dimension, it is not immediately clear that there are posets of unbounded dimension. The example below shows that this is the case.

Let $\mathbf{S}_n = (X, P)$ be the poset with ground set $X = \{a_1, a_2, \ldots, a_n\} \cup \{b_1, b_2, \ldots, b_n\}$, where for all $i, j \in \{1, 2, \ldots, n\}$, $a_i \| a_j$ and $b_i \| b_j$, and $a_i < b_j$ iff and only if $i \neq j$. The posets $\mathbf{S}_n$ are called the *standard examples*, and $\dim(\mathbf{S}_n) = n$.

To see that $\dim(\mathbf{S}_n) \leq n$, for each $i = 1, 2, \ldots, n$, let L_i be a linear extension of $\mathbf{S_n}$ with $a_i > b_i$ in L_i. It follows that $\{L_1, L_2, \ldots, L_n\}$ is a realizer. On the other hand, suppose $\dim(\mathbf{S}_n) = t$ and $\{L_1, L_2, \ldots, L_t\}$ is a realizer. Then, for each $i = 1, 2, \ldots, n$, there is an integer $j_i \in \{1, 2, \ldots, t\}$ so that $a_i > b_i$ in L_{j_i}. Now, for two integers i and k where $1 \leq i < k \leq n$, $a_k < b_i < a_i < b_k$ in L_{j_i}, and $j_i \neq j_k$. Hence $t \geq n$.

Tools from dimension theory

In this section we recall some facts from the dimension theory of finite posets. Again, the reader is referred to Trotter's monograph [58] for additional background and references.

A *critical pair* is a pair of incomparable elements (a, b) such that $x < b$ if $x < a$ and $y > a$ if $y > b$ for all $x, y \in \mathbf{P}$. A family of linear extensions $\mathcal{R}$ of P is a realizer of $\mathbf{P}$ if and only if each critical pair (a, b) is *reversed* in some linear extension $L \in \mathcal{R}$, i.e., $b < a$ in L. An incomparable min-max pair, i.e., a pair of incomparable elements (a, b) where a is a minimal element and b is a maximal element of $\mathbf{P}$, is always critical.

An *alternating cycle* in $\mathbf{P}$ is a sequence of critical pair $(a_0, b_0), \ldots, (a_k, b_k)$ such that $a_i < b_{(i+1 \mod (k+1))}$ for all $i = 0, \ldots, k$. A fundamental result is that $\dim(\mathbf{P}) \leq t$ if and only if there exists a t-coloring of the critical pairs of $\mathbf{P}$ such that no alternating cycle is monochromatic.

In the following example we illustrate how these facts can be combined to determine the dimension of a specific incidence order.

Example: Let M be the planar map of the complete graph K_4, and let $\mathbf{Q}_M$ be the poset on the vertices and faces of M ordered by inclusion. Every vertex has a single non-incident face, hence, there are four incomparable min-max pairs in $\mathbf{Q}_M$. These are all the critical pairs. Any two of these critical pairs form an alternating cycle. Therefore, the hypergraph of alternating cycles is again a K_4 and has chromatic number 4. This shows that $\dim(\mathbf{Q}_M) = 4$.

1.2 Reachability Oracles

Reachability is a classic algorithmic graph problem. It asks if there exists a path between two given vertices in the input graph. In some settings an actual path connecting the given vertices is also wanted. The question "Is there a path in from the vertex u to the vertex v in the graph $G = (V, E)$?" is called a *reachability query*. If the answer to the query is affirmative, we say that u *reaches* v (and v is *reached* by u).

In the simplest case, where only one query has to be answered, and no pre-computation is done, the reachability problem is solved by breadth-first search or depth-first search with u as starting vertex. This takes $O(|V| + |E|)$ time and uses $O(|V| + |E|)$ space. Reachability can, however, be solved using much less space: for directed graphs $O(\log |V|^2)$ space is sufficient [52], and recently Reingold proved that undirected reachability can be solved using just $O(\log |V|)$ space [49]. Reachability arises as a subproblem to be solved in many basic graph algorithms, e.g., in topological sorting, Ford-Fulkersons max-flow algorithm etc. (see [20] for further examples).

Suppose we instead want a data structure that can answer repeated reachability queries. Such a data structure is called a *reachability oracle*. For an undirected graph, it is easy to construct a reachability oracle that uses linear space and supports constant time queries: just label each vertex with the connected component it is in. In the rest of this section we will only discuss directed graphs. In fact, we only have to consider acyclic digraphs. Since each vertex in a strongly connected component (SCC) reaches and is reached by the same vertices, we can contract each SCC to a single vertex. Every acyclic digraph G induces a poset $\mathbf{P}_{V(G)}$ on the vertices $V(G)$ in G where $u < v$ iff there is a path from u to v. Hence, we can formulate the problem of constructing a reachability oracle for the graph G as the problem of efficiently representing $\mathbf{P}_{V(G)}$. For convenience, let n be the number of vertices in all graphs mentioned below, unless otherwise mentioned.

The traditional way to construct a reachability oracle is to compute the transitive closure of G and represent it as a matrix of size $n \times n$. Using such a matrix allows constant query time. However, the matrix representation requires $O(n^2)$ bits. Unfortunately, in general $\Omega(n^2)$ bits are required, as we will see in Chapter 2. However, if we restrict the input to special classes of graphs, the situation is entirely different.

Note that if the dimension of the poset $\mathbf{P}_{V(G)}$ is low, G has a small reachability oracle supporting fast query time. More precisely, if $\dim(\mathbf{P}_{V(G)}) = t$, then G has an $O(t)$ time $O(tn)$ space reachability oracle. Let $\{L_1, \ldots, L_t\}$ be a realizer of $\mathbf{P}_{V(G)}$. Label each element x in $\mathbf{P}_{V(G)}$ with a t-tuple $(x_1, \ldots, x_t)$, where x_i is the position of x in L_i. Now $x < y$ in $\mathbf{P}_{V(G)}$ iff $x_i < y_i$ for $i = 1, \ldots, t$, i.e., we use the standard dominance order.

Computational model The measure of space in the reachability oracle results above is *words*, unless explicitly stated otherwise. The word size is assumed to be just large enough to hold a vertex identifier, i.e., $O(\log n)$ bits for an n-vertex graph. Our computational model is thus the word RAM model [2]. Instructions like addition and multiplication operate on a constant number of words in a single time unit. This models what can be done in a standard programming language fairly well.

1.2.1 Reachability oracles for planar digraphs

Many of the previous oracles for reachability in planar digraphs work for distances as well. The first approach to reachability oracles for planar digraphs was to use the Lipton-Tarjan Separator Theorem [44], which says that for any planar graph G, there is a set of vertices S of size $O(\sqrt{n})$ such that no component of $G \setminus S$ contains more than $2|V|/3$ vertices. Arikati *et al.* [5] and Djidjev [22] constructed distance oracles with a space-time product of $O(n^2)$. More precisely, if their distance oracle uses space s, they can answer reachability queries in $O(n^2/s)$ time.

Djidjev [22] proved that with space $s \in [n^{4/3}, n^{3/2}]$, the query time can be improved to $O((n/\sqrt{s}) \log n)$ using the topology of planar graphs. For space $s = O(n^{4/3})$, this gives a query time of $O(n^{1/3} \log n)$ and a space-time product of $O(n^{5/3} \log n)$. Later, Chen and Xu [17] generalized this bound to space $s \in [n^{4/3}, n^2]$, achieving a distance query time of $O((n/\sqrt{s}) \log(n/\sqrt{s}) + \alpha(n))$.

Significant progress was made when Thorup 2001 presented a $O(n \log n)$ space reachability oracle with constant query time [57]. His reachability result can also be generalized to approximate distances. Before his result, no $o(n^2)$ bit oracle was known that could answer reachability queries in constant time. We will return to this result shortly, but first we discuss some special classes of planar graphs and distributed oracles.

A planar acyclic digraph G is *spherical s-t planar* if s is the only sink and t is the only source in G [55]. If we can add an edge from s to t in G while maintaining planarity, G is said to be *s-t planar*. It is well known that there are reachability oracles for s-t planar and s-t spherical graphs that use only linear space while still supporting constant query time [37, 55]. In the case of s-t planar graphs, this follows from the fact that the dimension of $\mathbf{P}_{V(G)}$ is 2.

Djidjev *et al.* [22] proved that if G is k-outerplanar, there is an $O(n \log n + k^2)$-space distance oracle with query time $O(\log n)$. For outerplanar graphs, this can be improved to oracle space $O(n \log \log n)$ and query time $O(\log \log n)$ [23].

1.2.2 Labeling schemes

A special kind of reachability oracle is the *labeling scheme* [47]. Here, the oracle distributes perfectly: each vertex v gets a label $D(v)$, and to answer the reachability query "Does u reach v?", we only have to inspect the labels $D(u)$ and $D(v)$. Such a labeling scheme is a called a *reachability labeling* of the vertices.

Gavoille *et al.* [34] proved that planar graphs admit a distance labeling with labels of size $O(\sqrt{n})$ supporting distance (and hence reachability) queries in time $O(\sqrt{n})$ using the Lipton-Tarjan Separator Theorem. They provide an almost matching lower-bound showing that even for undirected graphs, we need $\Omega(\sqrt{n})$-bit labels to support exact distance queries, no matter the time available to compute the distances.

1.2.3 Thorup's oracle construction

Thorup's oracle can also be implemented as a labeling scheme. The main idea is to construct a series of digraphs, such that any reachability query can be answered by considering a two of them, and such that each graph admits separators consisting of a constant number of directed paths.

Thorup first shows that any planar digraph G can be transformed into a series of digraphs $G_1, \ldots, G_\ell$ such that (i) the total sizes of the G_is is linear in the size of G, (ii) every vertex u has an index $\iota(u)$ such there is a path in G from u to v if and only if there is a path in $G_{\iota(u)}$ or $G_{\iota(u)-1}$ from u to v, (iii) each G_i has a spanning tree where each root path is the concatenation of two directed paths, and (iv) each G_i is an (undirected) minor of G. Thus, it suffices to find reachability oracles for the G_is.

Then, it is shown that in every such graph G_i, it is possible to find a set of vertices that induce a constant number of directed paths and whose removal separates the graph into connected component of balanced sizes in linear time using a component of the proof of the Lipton-Tarjan Separator Theorem.

It is then sufficient to store for each of the other vertices v the vertices in the separator paths that reaches or is reached by v. This can be done recursively.

1.2.4 Contributions

Now, what if we only care about a subset U of the vertices? This corresponds to a representation of the subposet $\mathbf{P}_U$ of $\mathbf{P}_{V(G)}$. It is fairly straightforward to change Thorup's labeling of the vertices in U to a reachability labeling with labels of size $O(\log |U|)$ in this case. However, we can do better.

If the set U is covered by f faces in a plane drawing of the input graph G, we prove that U has a reachability labeling with labels of size $O(f)$ and constant query time. With a different vertex weighting in the separator algorithm Thorup uses, we can change the recursion from the vertices in U to the faces that cover U. Using the $O(f)$-labeling as base case we thus get a reachability labeling of U with labels of size $O(\log f)$ supporting constant query time. For k-outerplanar digraphs (with $U = V$), this yields a reachability labeling with $O(\log k)$-sized labels and constant query time. For $k \in O(1)$, this is optimal.

1.3 Reachability substitutes

In the setting where only a subset of the vertices are interesting, it is possible to replace the input graph with a simpler graph, while preserving the existence of paths between interesting vertices. This problem arises as a subproblem in a dynamic reachability algorithm by Subramanian [54], but it is also an interesting combinatorial problem in its on right.

More precisely, we consider the following network design problem. Let $G = (V, E)$ be a directed graph with $n = |V|$ vertices and let $U \subseteq V$ be a set of κ vertices in G which are designated *interesting*. A *reachability substitute* (G, U) is a digraph $H = (V', E')$ such that $U \subseteq V'$ and for any two interesting vertices $u, v \in U$, there is a path from u to v in H iff there is a path from u to v in G. For convenience, we sometimes speak of H as a reachability substitute of the graph G when $U = V$. Of course, G itself is a substitute for (G, U) for any $U \subseteq V$. The problem we are interested in is that of finding a *small* reachability substitute, i.e., one that minimizes $|V'| + |E'|$.

We can also frase this problem as a poset problem. Construct an acyclic graph G' by contracting each strongly connected component. A substitute for

G' can now be turned into a substitute for G by replacing each contracted vertex with a simple cycle on the interesting vertices in the corresponding strongly connected component in G. Clearly, this preserves optimality. Hence, we only have to consider acyclic digraphs.

The input of the reachability substitute problem is the following in poset terms: a poset $\mathbf{P}_U$ is given together with its superposet $\mathbf{P}_{V(G)}$. A reachability substitute is a different superposet of $\mathbf{P}_U$. The task is now to find a superposet with cover graph of minimum size.

Aside from the complexity of finding a small reachability substitute, the structural question of lower and upper bounds on the size of substitutes is interesting. Trivially, every input pair (G, U) has a substitute of size $O(|U|^2)$.

If all vertices are interesting ($U = V$) and one also requires $V' = V$ and $E' \subseteq E$, the problem is called *Minimum Equivalent Graph (MEG)*. It is NP-hard and can be approximated within a constant factor in polynomial time [40]. If only $V' = V$ is demanded while E' is allowed to contain edges that are not in E, the problem can be solved in polynomial time [1, 36, 40]: Compute the transitive reduction of the acyclic digraph G' corresponding to the input graph G. Note that the reachability substitute problem treated in this thesis, besides allowing $U \subset V$, removes both requirements of MEG: V' does not need to be contained in V and E' does not need to be a subset of E.

1.3.1 Steiner graphs and spanners

A Steiner graph $G' = (V', E')$ of a digraph $G = (V, E)$ is an edge-weighted digraph such that $V \subseteq V'$ and the distances d_G and $d_{G'}$ satisfy $d_{G'}(u, v) \geq d_G(u, v)$ for all vertices $u, v \in V$. Bollobás, Coppersmith and Elkin [12] studied Steiner graphs that preserves large distances. A *Steiner d-preserver* is a Steiner graph G' such that $d_{G'}(u, v) = d_G(u, v)$ if $d_G(u, v) \geq d$. Hence, a Steiner 1-preserver is a reachability substitute of G. Moreover, Bollobás, Coppersmith and Elkin proved that every graph has a Steiner 1-preserver of size $O(n^2/\log n)$, so each pair (G, U) has a reachability substitute of size $O(|U|^2/\log |U|)$.

A substitute graph that not only preserves the existence of paths but also approximates their lengths is called a *spanner*. Several spanner constructions, with different size and approximation guarantees, are known, though most of the existing work on spanners is about undirected graphs [4, 9, 10, 18, 25–27, 48, 51]. Recently, Coppersmith and Elkin [19] introduced a new variant of spanners which is more closely related to the problem considered here. In a special case they try to find a small subgraph of a given (undirected) graph, also with a subset of interesting vertices, such that in the subgraph we have the same distances between interesting vertices as in the original graph. They show that there always exists such a subgraph with $O(n)$ edges if the number of interesting vertices is $O(n^{1/4})$. Even more recently, Klein [41] showed that undirected planar graphs have small subset-spanners. That is, he proved that for a planar graph G with weights on the edges and a subset U of the vertices designated as interesting, for every constant $\epsilon > 0$, there is a subgraph H of G such that (1) H contains all vertices of U, (2) the distance between every two vertices in H is at most $1 + \epsilon$ times their distance in G, and (3) the total weight of H is at most a constant times the weight of a Steiner tree for U in G.

1.3.2 Reachability substitutes for planar digraphs

Similar to reachability oracles section, our main focus is on planar inputs — the output substitutes need not be planar, though. Subramanian [54] showed that if all vertices of U lie on a constant number of faces of a planar embedding then there is a solution of size $O(|U| \log |U|)$, which can be found in $O(n \log n)$ time. That algorithm was designed as a component of an algorithm for dynamic reachability in planar graphs; the graph is recursively partitioned with small separators, where the separator vertices become the interesting vertices. Eppstein *et al.* [28] generalized this approach and obtained fully dynamic algorithms for several problems on planar graph. In addition, Klein and Subramanian [42] showed that the algorithm of [54] can be modified to construct a substitute graph that not only represents the existence of paths between the interesting vertices but also approximates the lengths of shortest paths.

1.3.3 Contributions

By a counting argument, we prove that at most a fraction $o(1)$ of the pairs (G, U) of graphs and sets of interesting vertices have reachability substitutes of size $o(|U|^2/\log |U|)$. From Bollobás, Coppersmith and Elkin's result on Steiner 1-preservers, we know that this bound is tight.

We show how to change Thorup's oracle construction to produce a reachability substitute of size $O(|U| \log |U|)$ for planar input graphs. The substitutes that we construct are not planar however — we show that this is unavoidable. We exhibit an example of a graph with no planar reachability substitute of size $o(|U|^2)$.

The combination of the lower bound on the size of substitutes for general graphs and the upper bound on the size of reachability substitutes for planar graphs implies that only a tiny minority of digraphs have a planar substitute. We briefly discuss the possibility of characterizing such graphs.

1.4 Dimension

The dimension is a widely studied parameter of posets. Since its introduction by Dushnik and Miller [24] in 1941, it has moved into the core of combinatorics. There are close connections and analogies with the chromatic number of graphs and hypergraphs. From the applications point of view, dimension is attractive because low dimension warrants a compact representation of the poset, as we saw in Section 1.2. Trotter [58, 59] provides a more extensive introduction to the area.

Yannakakis showed that for $t \geq 4$, it is NP-hard to decide if $\dim(\mathbf{P}) \leq t$ for height 2 posets [60]. For posets $\mathbf{P}$ of height 3 or more, the problem becomes NP-hard already for $t = 3$. On the other hand, there are fast algorithms to test whether a poset $\mathbf{P}$ is of dimension 2, see e.g. [46]. The complexity of deciding if the dimension of a height 2 poset is at most 3 is a major open problem in dimension theory.

1.4.1 Schnyder's Theorem and the dimension of graphs

There are many connections between posets and graphs [59]. One of most studied is the dimension of the posets defined by the incidence structures in graphs.

The *vertex-edge poset* (or *incidence poset*) $\mathbf{P}_G$ of a graph G is the poset on the vertices of edges of G ordered by inclusion. More precisely, let $G = (V, E)$ be a graph. Then the vertex-face poset $\mathbf{P}_G = (X, P)$, where $X = V \cup E$ and $(v, e) \in P$ if and only if $v \in V$, $e \in E$ and v is an endpoint of e.

The study of of the dimension of the vertex-edge poset of a graph is justified by Schnyder's celebrated theorem.

Theorem 1.1 (Schnyder [53])**.** *A graph G is planar if and only if* $\dim(\mathbf{P}_G) \leq 3$.

Felsner and Trotter [31] defined the closely related concepts of a realizer and dimension of a graph $G = (V, E)$. A realizer of G is a nonempty family $\mathcal{R}$ of linear orders on V such that for every edge $e \in E$ and every vertex $v \in V \setminus e$ there is some $L \in \mathcal{R}$ such that $v > w$ in L for all vertices $w \in e$. As in the case of posets, the dimension of G, $\dim(G)$, is the minimum cardinality of a realizer of G.

We can formulate Schnyder's Theorem in terms of graph dimension instead of poset dimension. To avoid uninteresting special cases, we assume that we are dealing only with connected graphs with 3 or more vertices (all other graphs are planar and have vertex-edge posets of dimension at most 3 anyway). If we insert the edges of a graph G as low as possible in each linear extension in a realizer $\mathcal{R} = \{L_1, \dots, L_t\}$ of G, we get a set of linear extensions $\mathcal{R}' = \{L'_1, \dots, L'_t\}$ of $\mathbf{P}_G$. The only critical pairs not reversed in some $L'_i \in \mathcal{R}'$ are of the form (w, v), where v, w are end points of the same edge and w has degree 1. Hence, it is not hard to see that $\dim(\mathbf{P}_G) \leq 3$ if and only if $\dim(G) \leq 3$.

The dimension of a graph can be used to characterize other classes of graphs. We can refine the definition of dimension as follows: we say that the dimension of a graph G is $[t-1 \updownarrow t]$ if $\dim(G) > t-1$ and G has a realizer $\{L_1, L_2, \dots, L_t\}$ with $L_{t-1} = L_t^d$ [31], where L_t^d is the dual poset of L_t, i.e., the poset on the same ground set as L_t such that $x < y$ in L_t^d if and only if $x > y$ in L_t. Using Schnyder's Theorem, Felsner and Trotter [31] proved that a graph is outerplanar if and only if its dimension is at most $[2 \updownarrow 3]$.

The proof that G is planar if $\dim(G) \leq 3$, actually due to Babai and Duffus [8], is relatively easy. To prove the other direction, Schnyder developed a more involved combinatorial machinery with several lemmas of independent interest. For instance, the classic results that planar graphs have arboricity at most 3, i.e., that the edges of every planar graph can be partitioned into 3 disjoint forests, follows from Schnyder's oriented coloring of the edges in triangulations. These *Schnyder woods* are families of three trees such that each vertex has one outgoing edge in each tree.

1.4.2 Planar maps

Schnyder's theorem can be extended from planar graphs to planar maps. Similar to the vertex-edge poset of graph, the vertex-edge-face poset $\mathbf{P}_M$ of a planar map M is the poset on the vertices, edges and faces of M ordered by inclusion. The vertex-face poset $\mathbf{Q}_M$ of M is the subposet of $\mathbf{P}_M$ induced by the vertices and faces of M.

The theorems by Brightwell and Trotter cited below are the starting point for our investigation into the order dimension of planar maps.

Theorem 1.2 (Brightwell and Trotter [15])**.** *Let M be a planar map. Then* $\dim(\mathbf{P}_M) \leq 4$*.*

Theorem 1.3 (Brightwell and Trotter [14])**.** *For every 3-connected planar map M,* $\dim(\mathbf{P}_M) = \dim(\mathbf{Q}_M) = 4$*. Furthermore, if one face or vertex is removed from $\mathbf{P}_M$,* $\dim(\mathbf{P}_M) = \dim(\mathbf{Q}_M) = 3$*.*

Brightwell and Trotter [15] have also presented a complete characterization of the planar maps with $\dim(\mathbf{P}_M) = 2$.

1.4.3 Contributions

Most maps have dimension at least 3. For $\dim(\mathbf{P}_M) \geq 3$ it is sufficient that M has a vertex of degree 3. Motivated by this we approach the problems of characterizing the maps M with $\dim(\mathbf{P}_M) \leq 3$ and the maps with $\dim(\mathbf{Q}_M) \leq 3$. These characterization problems have earlier been posed by Brightwell and Trotter [15].

We prove that for $\dim(\mathbf{Q}_M) \leq 3$ it is necessary that M is K_4-subdivision free. This is done by showing that a map containing a K_4-subdivision must contain the vertex-face poset of some 3-connected planar map as a subposet. For $\dim(\mathbf{P}_M) \leq 3$ an additional necessary condition is that both M and M^* are $K_{2,3}$-subdivision free. We show this by establishing that each path in a $K_{2,3}$-subdivision induces a fence poset, i.e., a poset with elements $\{x_1, \ldots, x_k$ such that $x_1 < x_2 > x_3 < x_4 > \ldots < x_{k-1} > x_k$, and that such a triple of fences force $\dim(\mathbf{P}_M) > 3$. Together, these results mean that if $\dim(\mathbf{P}_M) \leq 3$, then both M and M^* are outerplanar.

We then study the simplest class of maps M such that M and M^* are outerplanar, which we call *path-like*. For maximal path-like maps we prove that $\dim(\mathbf{P}_M) \leq 3$ is equivalent to the existence of a special oriented coloring of the interior edges and characterize the path-like maps which admit such a coloring. The characterization is turned into a linear time algorithm that generates a 3-realizer, i.e., three linear extensions whose intersection is $\mathbf{P}_M$, or returns the information that $\dim(\mathbf{P}_M) \geq 4$.

Finally, we prove that if M is 2-connected and M and M^* are outerplanar, then $\dim(\mathbf{Q}_M) \leq 3$. We also present a strongly outerplanar map with a vertex-face poset of dimension 4. No such map was known before. The example, a maximal outerplanar graph with 21 vertices, is quite large. We provide some arguments which indicate that our example is not far from being as small as possible. Given these results we raise the question of if it is NP-hard to decide if $\dim(\mathbf{Q}_M)$ already for strongly outerplanar maps M.

1.5 Large treewidth

The final theme mentioned in the beginning of this chapter is special cases of NP-hard problems. In the last section, we discussed special cases of the NP-hard poset dimension problem. In this section we consider graph problems.

One of the most successful parameterizations of graphs is that of *treewidth*. While the formal definition is deferred to Chapter 5, graphs of treewidth k, also known as *partial k-trees*, are graphs that admit a tree-like structure, known as *tree-decomposition* of *width k*.

Many NP-hard problems have been shown to be solvable in polynomial time, or even linear time, for graphs with treewidth k [6, 7, 56]. For some of these problems, polynomial time algorithms exists possible even for graphs of treewidth $O(\log n)$ or $O(\log n/\log\log n)$ [7, 56].

A standard example of a problem solvable in graphs of treewidth $O(\log n)$ is the maximum independent set (MIS) problem [7], which is that of finding a maximum cardinality set of pairwise non-adjacent vertices. For general graphs, the best polynomial-time approximation ratios known for MIS is $n\,(\log\log n)^2/\log^3 n$ [29]. On the other hand, it is known that unless NP $\subseteq$ ZPTIME$(2^{(\log n)^{O(1)}})$, no polynomial-time algorithm can achieve an approximation guarantee of $n^{1-O(1/(\log n)^{\gamma})}$ for some constant γ [39].

We study the approximability of some of the aforementioned NP-hard problems, mainly considering graphs of treewidth $k = \omega(\log n)$. We focus our study on MIS, deriving further applications of our method by extensions of that given for MIS.

Better approximation bounds for MIS are achievable for special classes of graphs. A class that properly contains the graphs of treewidth at most k is that of *k-inductive graphs*. A graph is said to be k-inductive if there is an ordering of its vertices so that each vertex has at most k higher-numbered neighbors. If such an ordering exists, it can be found by iteratively choosing and removing any vertex of minimum degree in the remaining graph. The best approximation known for MIS in k-inductive graphs is $O(k\log\log k/\log k)$ [35].

1.5.1 Contributions

We develop a generic scheme for approximation algorithms for maximum independent set and other NP-hard graph optimization problems in graphs with treewidth $k = \omega(\log n)$. This scheme leads to polynomial-time algorithms with approximation ratio $\ell/\log n$ when a tree-decomposition of width $\ell = \Omega(\log n)$ is given. For MIS, this improves the previously best known bound by a factor $\log\log k$ if $\ell = k$.

Our scheme can be applied to any problem of finding a maximum induced subgraph with hereditary property Π and any problem of finding a minimum partition into induced subgraphs with hereditary property Π provided that for graphs given with tree-decompositions of logarithmic or near logarithmic width it can be solved exactly in polynomial time. All approximation factors achieved are the best known for the aforementioned problems for graphs of superlogarithmic treewidth. See Table 5.1 for a comparison with previous results.

In case a tree-decomposition of width $\ell = k$ is not given, the approximation achieved by our method increases by a factor of $O(\sqrt{\log k})$.

1.6 Outline of the thesis

In Chapter 2 we discuss reachability oracles and prove that planar graphs with U interesting vertices such that f faces covers U admit reachability labelings with labels of size $O(\log f)$ and constant query time. This chapter is based on the paper

- Johan Nilsson. Improved reachability oracles for planar digraphs. In preparation.

Chapter 3 contains the discussion of reachability substitutes. It is based on the paper

- Irit Katriel, Martin Kutz, Johan Nilsson and Martin Skutella. Reachability substitutes for planar digraphs. 2006. Submitted to *Journal of Graph Algorithms and Applications*. Under revision.

In Chapter 4 we investigate the order dimension of planar maps. This chapter is based on

- Stefan Felsner and Johan Nilsson. On the order dimension of outerplanar maps. 2007. Submitted to the journal *Combinatorics, Probability and Computing*. An extended abstract of this paper has been submitted to the 19th ACM-SIAM Symposium on Discrete Algorithms.

Finally, in Chapter 5 we provide a scheme for approximation algorithms for optimization problems in graphs with superlogarithmic treewidth based on the paper

- Artur Czumaj, Andrzej Lingas, Magnús M. Halldórsson and Johan Nilsson. Approximation algorithms for optimization problems in graphs with superlogarithmic treewidth. *Information Processing Letters* 94(2):49–53 (2005).

Chapter 2

Reachability oracles

One of the most fundamental algorithmic graph problems is determining if there is a path from a vertex u to another vertex v in a directed graph G. If such a path exists, v is said to be reachable from u. Suppose we want a data structure that can answer queries of the type "Is v reachable from u?". Such a structure is called a *reachability oracle*. For an undirected graph, it is easy to construct a reachability oracle that uses linear space and supports constant time queries: just label each vertex with the connected component it is in.

The traditional way to construct a reachability oracle is to find the transitive closure of G and represent it as a $n \times n$ matrix, where $n = |V(G)|$. This allows us to answer reachability queries in constant time, but the matrix representation requires $O(n^2)$ bits.

Can we do much better? Unfortunately, this is not the case. A very simple argument shows that $\Omega(n^2)$ bits are necessary: Consider the set of bipartite n-vertex digraphs with $n/2$ vertices in each of bipartitions A and B, such that all edges are directed from A to B. Now, for each of the $n^2/4$ pairs (a, b) of vertices $a \in A$, $b \in B$, the graph where (a, b) is an edge and the graph where it is not an edge must have different reachability oracles. Hence, at least $n^2/4$ bits are necessary.

However, if we restrict the input to special classes of graphs, the situation is entirely different. For planar digraphs, Thorup has shown that it is possible to construct constant time, $O(n \log n)$ space reachability oracles for planar digraphs [57]. Below, we will describe Thorup's construction in more detail. Note that the measure of space in the results above, as well as throughout the chapter, will be *words*, unless explicitly stated otherwise. The word size is assumed to be just large enough to hold a vertex identifier, i.e., $O(\log n)$ bits.

Thorup's oracle can be distributed perfectly as labels on the vertices: each vertex v is given a label $D(v)$ such that we can test if u reaches v by just inspecting the labels $D(u)$ and $D(v)$. We call such a labeling scheme a *reachability labeling*. Thorup's reachability labeling uses labels of size $O(\log n)$ and supports constant query time.

Note that a reachability oracle is the same as a representation of the poset $\mathbf{P}_{V(G)}$, i.e., the poset on $V(G)$ where $u < v$ iff u reaches v. As we saw in the introduction, if $\mathbf{P}_{V(G)}$ has dimension t, $V(G)$ has a reachability labeling with labels of size $O(t)$ and query time $O(t)$. The dimension does not help much for planar digraphs, though. There are planar digraphs G such that the poset $\mathbf{P}_{V(G)}$ contains the standard example $\mathbf{S}_r$ for $r \in O(\sqrt{|V(G)|})$ as a subposet (see Figure 2.1 for an example), so $\dim(\mathbf{P}_{V(G)}) = O(\sqrt{n})$.

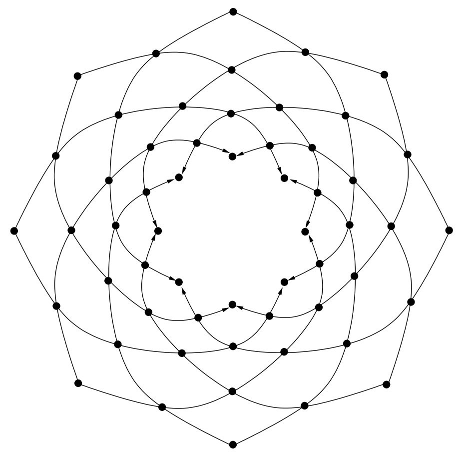

Figure 2.1: A planar digraph G with 56 vertices such that $\mathbf{S}_8$ is a subposet of $\mathbf{P}_{V(G)}$.

2.0.1 Contributions

If we consider the slightly different problem, where we only are interested in queries "Is v reachable from u?" for some $u, v \in U \subseteq V(G)$, we can find a better labeling. Suppose there are f faces in a plane drawing of G such that every vertex of U is in one of these faces. Then there is a reachability labeling with labels of size $O(\log f)$. This is optimal for constant f. In particular, it is optimal for k-outerplanar digraphs, where k is constant.

2.1 Thorup's oracle construction

We begin with a brief sketch of the relevant details of Thorup's construction of reachability oracles [57].

The main idea is to construct a series of digraphs, such that any reachability query can be answered by considering a constant number of them, and each graph admits separators consisting of a constant number of directed paths. Thorup first shows that any planar digraph G can be transformed into a series of digraphs $G_1, \dots, G_\ell$ such that

- The total sizes of the G_is is linear in the size of G.
- Every vertex u has an index $\iota(u)$ such there is a path in G from u to v if and only if there is a path in $G_{\iota(u)}$ or $G_{\iota(u)-1}$ from u to v.
- Each G_i is a 2-layered digraph, i.e., a digraph that has a spanning tree where each root path is the concatenation of two directed paths.

- Each G_i is an (undirected) minor of G.

Thus, it suffices to find reachability oracles for the G_is.

The G_is are constructed as follows. First, the vertices of G are partitioned into layers. We start with an arbitrary vertex v_0. The layer L_0 consists of the vertices reachable from v_0. Let $L_{<i}$ denote $\cup_{j<i} L_j$. The layer L_i then contains the vertices in $V(G) \setminus L_{<i}$ reachable from a vertex in $L_{<i}$ when i is even, and the vertices in $V(G) \setminus L_{<i}$ that reach a vertex in $L_{<i}$ when i is odd (see Figure 2.2). The graph G_i is constructed from $G[L_{<i+2}]$ contracting all vertices in $L_{<i}$ into a single vertex r_0 for $i > 0$. For G_0, $r_0 = v_0$.

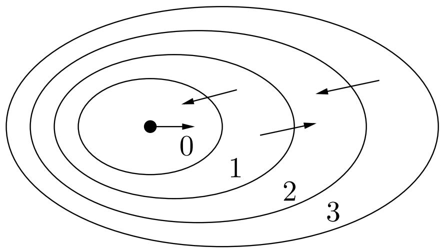

Figure 2.2: The first four layers.

Then, it is shown that in every 2-layered digraph, it is possible to find a set of vertices that induce a constant number of directed paths and whose removal separates the graph into maximal connected component of balanced sizes in linear time. More precisely, the following lemma (which essentially appears in [44]) is proved.

Lemma 2.1 (Thorup [57], Lemma 2.3)**.** *Given a vertex-weighted undirected planar graph H with a rooted spanning tree T, in linear time, we can find three vertices u, v and w such that if we remove the vertices in the paths $T(u)$, $T(v)$ and $T(w)$ from u, v and w to the root of T, no connected component of $H \setminus V(T(u) \cup T(v) \cup T(w))$ have weight more than $1/2$ of the weight of H.*

2.1.1 Recursive framed separators

In the following sections we show how to use Lemma 2.1 for a reachability labeling of each 2-layered graph G_i.

We recursively separate G_i using Lemma 2.1. With each subgraph $H \subseteq G_i$ we will pass down a set R of root paths of T that separates H from the rest of the graph G_i. This set of root paths R is called the *frame* of H, and H is a component in $G_i \setminus V(R)$. Hence, if S separates u from v in H, $R \cup S$ separates u from v in G_i. For each directed path q induced by a subset of $V(R) \cup V(S)$, we will identify the connections between H and q over G_i, i.e., the vertex $v \in V(H)$ connects to (from) the first (last) vertex in q that v reaches in G_i. We store the position of these vertices in q as $\text{to}_v[q]$ and $\text{from}_v[q]$.

If R and S are of constant size, there will only be a constant number of separator and frame paths to query to check reachability from u to w. The frame R is a set of root paths in T whose cardinality we want to minimize, so we can assume that each path starts in a leaf of the subtree $T[V(R)] \subseteq T$ induced by $V(R)$. For convenience, we call he leaves of the tree $T[V(R)]$ the leaves of R.

2.1.2 Bounding the number of frame paths

In order to keep the frames of constant size, we alternate between two types of recursive calls of Lemma 2.1 with different vertex weightings: subgraph reducing calls and frame reducing calls. Let $H + R$ denote H and R together with all edges from G_i between H and R. In the subgraph reducing recursion, we pick a separator $S = \{T(u), T(v), T(w)\}$ such that no component of $(H + R) \setminus V(S)$ contains more than half of $V(H)$. This is done by giving the vertices in H weight 1 and vertices in R weight 0 when applying Lemma 2.1 to $H + R$.

In a frame reducing recursion, we pick S so that no component of $(H + R) \setminus V(S)$ contains more than half the leaves of R. Thus, in a frame reducing recursion these leaves are given weight 1, whereas all other vertices in $H + R$ are given weight 0. Since every other recursive call halves the weight of the subgraph H, the recursion will have depth $O(\log |V(G_i)|)$.

We will now consider the number of root paths in the frames passed down to the recursions on the components of $H\ V(S)$, starting with a subgraph reducing recursive call. If the original frame R has α root paths, then trivially, each component receives a frame with at most $\alpha' = \alpha + 3$ root paths; those from R plus the three in S. In a frame reducing recursive call, we only pass down S and the root paths in R who starts in $H \setminus V(S)$ as a frame. Since each frame path starts in a leaf, at most $\alpha'/2 + 3 = \alpha + 4.5$ frame paths will be passed down. The graph G_i has no frame, so there are no more than 12 frame paths at any time during the recursion. Each of these frame paths corresponds to two directed paths in G_i.

2.1.3 Indexing with frames

Every vertex v participates in all recursive calls in the recursion tree along the path from the root of the recursion tree to the call where it is added to the separator S. This last call is called the *final call* of v. In each ancestor call of the final call of v, the vertex v enumerates both the directed paths in the frame and the directed paths in the separator. This means that the same directed path might be numbered several times; first as a separator, and later as a frame. The enumeration is done such that each vertex in a call gives the path q the same number.

The *separation number* of a call is the last number used for enumerating directed paths for that call. Now, let C be the nearest common ancestor of the final calls of u and w, and let s be the separation number of C. If p is the separation number of the parent of C, then $\{p + 1, ..., s\}$ will be the indices of directed paths of the frame and the separator of C, so u reaches w in G_i if and only if there exists a $q \in \{p + 1, ..., s\}$ such that $\mathrm{to}_u[q] \leq \mathrm{from}_w[q]$.

2.1.4 Distributing the oracle

The reachability oracle constructed so far can be distributed as a labeling scheme. With each vertex v we associate a label $D(v)$ of size $O(\log n)$, such that given only $D(u)$ and $D(w)$, we can determine if u reaches w.

The first parts of the label for v are the $O(\log n)$-space tables from_v and to_v. In able to distribute these parts, we cannot store the separation numbers globally with the calls in the recursion tree anymore. Instead, for each vertex v and depth d, we store the separation number $s_v[d]$ of the depth d ancestor of the final call of v in the recursion tree. Then we can just use a labeling scheme for depths of nearest common ancestors. Alstrup *et al.* [3] proved that such a labeling with labels of size $O(\log n)$ bits and constant query time can be computed in time $O(n)$, where n is the number of vertices in the tree.

Theorem 2.2 (Thorup [57])**.** *Let G be a planar n-vertex digraph. Then we can, in $O(n \log n)$ time, compute a reachability labeling of $V(G)$ with labels of size $O(\log n)$ and constant query time.*

Now, suppose we are only interested in a subset $U \subseteq V(G)$ of the vertices. We then want a labeling of the vertices in U such that we can answer reachability queries about two vertices $u, v \in U$ by just inspecting the labels of u and v. It is fairly straightforward to modify Thorup's construction so that each vertex in U gets a label of size $O(\log |U|)$ while still having constant query time. In each subgraph reducing recursion, instead of giving all the vertices in $V(H)$ weight 1, we only give vertices in U weight 1 (all other vertices get weight 0). We then only consider connections between vertices in U and the directed paths in the frames and separators.

2.2 An improved labeling scheme

We next prove that Thorup's labeling can be improved if the vertices are contained in the input graph.

Suppose we are interested in a set $U \subseteq V(G)$ of vertices in the planar digraph G. Let $\mathcal{F}$ be a set of faces in a plane drawing of G. Recall that $\mathcal{F}$ covers U if for each vertex $u \in U$ there exists some face $F \in \mathcal{F}$ such that $u \in F$. It turns out if we are given a set $\mathcal{F}$ that covers U such that $|\mathcal{F}| \in o(|U|)$, we can use smaller labels.

Theorem 2.3. *Let G be planar n-vertex digraph with a set of U interesting vertices given with a plane drawing of G, such that U is covered by the set $\mathcal{F}$ of faces. Then we can, in $O(\min\{n \log n, n|U|\})$ time, compute a reachability labeling of U with labels of size $O(\log |\mathcal{F}|)$ and constant query time.*

Contracting the vertices in $L_{<i}$ does not change the incidences between faces in $\mathcal{F}$ and vertices in L_i and L_{i+1}, so still can consider each 2-layered graph G_i separately.

We change the recursion so that each subgraph reducing recursive call roughly halves the number of faces in $\mathcal{F}$ that contains a vertex in U in each component. Then we show how to construct a reachability labeling with labels of size $O(|\mathcal{F}|)$ and use that as a base case of our recursion.

We start by constructing a new graph $G_{i,\mathcal{F}}$ by adding a new vertex v_F in each face $F \in \mathcal{F}$ and connecting it to all the other vertices $v \in F$ by inserting new edges $\{v, v_F\}$. These new vertices are called *face vertices*. We then augment the spanning tree T to a spanning tree of $G_{i,\mathcal{F}}$ by adding one of these new edges. During the recursion, we will split these vertices in a way such that in the end, each component H will contain a constant number of face vertices connected to the vertices in $U \cap V(H)$.

Throughout the recursion, we will maintain the invariant that each face vertex is a leaf in the spanning tree and that every vertex in U that is not in a separator or a frame is connected to a face vertex corresponding to a face in $\mathcal{F}$.

This is immediately true after each application of Lemma 2.1, unless a face vertex is in the separator. Now, suppose v_F is in the separator S for some face $F \in \mathcal{F}$ after we have applied Lemma 2.1 to the graph H during the recursion.

We add one copy of v_F to each component H' of $H \setminus V(S)$ where $V(H') \cap F \neq \emptyset$ and connect it to the vertices in $V(H') \cap F$ (see Figure 2.3). We then augment the spanning tree of H' by adding one of the new edges like before. Since every face vertex is a leaf of the spanning tree, at most 3 face vertices can be split in this way.

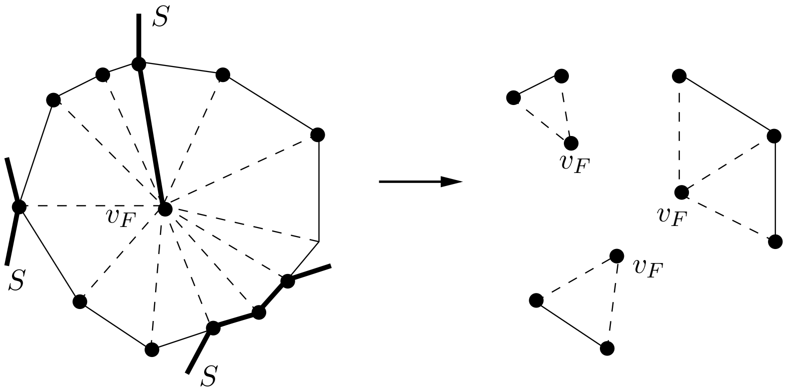

Figure 2.3: The face vertex v_F is split if it is contained in the separator.

In a subgraph reducing recursive call, we give weight 1 to the face vertices and weight 0 to all the other vertices. If H contains f face vertices before applying Lemma 2.1, no augmented component of $H \setminus V(S)$ contains more than $f/2 + 3$ face vertices after the face vertices are split. In a frame reducing recursive call the weighting is the same as before. The number of face vertices in an augmented component of $H \setminus V(S)$ can hence be at most $f + 3$. Hence, after two recursive calls the number of face vertices in a component goes from f to at most $f/2 + 6$.

For sufficiently large f, $f/2 + 6$ is a constant fraction of f. Hence, after $O(\log |\mathcal{F}|)$ recursive calls, each component contains a constant number of face vertices. This means that for each component H, $U \cap V(H)$ is covered by a constant number of faces in $\mathcal{F}$.

2.2.1 Labels of size $O(|\mathcal{F}|)$

As base case for our recursion, we need a compact reachability labeling when $|\mathcal{F}|$ is constant. We start with the case $|\mathcal{F}| = 1$.

Lemma 2.4. *Let G be planar graph with a set $U \subseteq V(G)$ of interesting vertices given with a plane drawing of G such that all vertices in U are in a single face F. Then there is a reachability labeling of U using labels of constant size with constant query time.*

Proof. We assign each vertex $u \in U$ a natural number $C(u)$ by giving an arbitrary start vertex x the number $C(x) = 1$ and giving the other vertices increasing numbers going clockwise around the boundary of F from x.

The label of each vertex v will consist of 5 parts. The first part is the number $C(v)$. We then have two labels each for the cases $C(u) < C(v)$ and $C(u) > C(v)$, where u is the other vertex involved in the reachability query: one label for paths from u to v (the incoming label) and one label for paths from v to u (the outgoing label). We only have to consider paths from u to v where $C(u) < C(v)$; the other case is symmetric.

First, we note the following useful fact.

Fact A. Let u, v, w, s and t be vertices in U such that u and v both reach the vertex t and w reaches the vertex s. If $C(v) < C(w) < C(u) < C(s)$, then w reaches t.
Proof. The path between w and s must cross either the path between v and t (if $C(s) > C(t)$) or the path between u and t (if $C(s) < C(t)$) (see Figure 2.4). $\triangle$

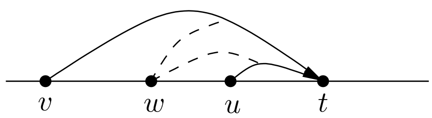

Figure 2.4: The vertex w cannot reach any vertex to the right of u without reaching t.

We then construct a graph T on the vertices in U as follows: $(u, v) \in E(T)$ if u and v reach a common vertex, $C(u) > C(v)$ and for every other vertex w that reaches a common vertex with u, $C(u) > C(w)$ implies $C(v) > C(w)$. Clearly, T is acyclic and every vertex has at most one outgoing edge in T. We construct a new weakly connected graph T' from T by adding a new vertex r and edges (s, r) for all vertices $s \in V(T)$ with outdegree 0. It is easy to see that the underlying undirected graph T'' of T' is a tree. We choose r as root of this tree.

Claim P. The set of vertices $S(t)$ that reach a vertex t in G induces a path in T'.
Proof. Let u, v be the vertices in $S(t)$ such that $C(u) > C(v)$ and such that for all $w \in S(t)$, $C(u) > C(w)$ implies $C(v) > C(w)$. Suppose $C(v) < C(z) < C(u)$ and z and u both reach the vertex $s \neq t$ for some vertex $z \notin S(t)$. Then, by

Fact A, z must reach t, which is a contradiction. Hence, (u, v) is an edge in T'. The claim follows by induction. △

Hence, every set $S(t)$ induces a connected subgraph of root path in T''. We can now give each vertex v an incoming and outgoing label: the outgoing label is the pre- and post-order numbers x and y of v in T'', and the incoming label is the pre- and post-order numbers x_1, y_1 of the vertex in $S(v)$ that is highest in T'' and the pre- and post-order numbers x_2, y_2 of the vertex in $S(v)$ that is lowest in T''. A query 'u reaches v?' is then answered by checking if $x_1(v) \leq x(u) \leq x_2(v)$ and $y_2(v) \leq y(u) \leq y_2(v)$. □

Since we already have a labeling for paths between vertices in the same face, we only need to add labels for paths between vertices in different faces when $|\mathcal{F}| \leq 2$.

Lemma 2.5. *Let G be planar graph with a set $U \subseteq V(G)$ of interesting vertices given with a plane drawing of G such that all vertices in U are in the two faces F_1 and F_2. Then there is a labeling of the vertices in U using labels of constant size such that we can answer queries "is v reachable from u?" for $u \in F_1$ and $v \in F_2$ in constant time.*

Proof. W.l.o.g. we assume that each vertex in $U \cap F_1$ reach some vertex in $U \cap F_2$. We start by constructing a directed cycle C on the vertices in $U \cap F_1$ by adding edges directed from each vertex in U to the nearest clockwise vertex in U.

Claim P. The set of vertices $S(t)$ in $U \cap F_1$ that reaches a vertex $t \in U \cap F_2$ induces a path in C.

Proof. Suppose not. Then there are vertices $u, v, w, z \in F_1 \cap U$ and a vertex $t \in F_2 \cap U$, such that the vertices come in the order u, v, w, z in C, $u, w \in S(t)$ and $v, z \notin S(t)$ (see Figure 2.5)). Now, the paths from u and w to t together with the boundary cycle in F_1 delimits a region in the plane such that one of v and z is inside the region, and all the vertices in $(U \cap F_2) \setminus \{t\}$ is outside the region. W.l.o.g. assume v is inside the region. But v reaches some vertex $s \in (U \cap F_2) \setminus \{t\}$. The path to s must hence intersect either the path from to u to t or the path from w to t, so v also reaches t; a contradiction. △

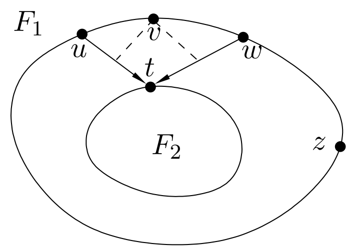

Figure 2.5: The vertex v cannot reach any vertex in $F_2 \setminus \{t\}$ without reaching t.

Now, we can just number the vertices in $U \cap F_1$, like in the proof of Lemma 2.4, giving each vertex u a natural number $C(u)$. Each vertex $u \in U \cap F_1$

gets the number $C(u)$ as its label, and each vertex $v \in F_2 \cap U$ gets $C(v_1)$ and $C(v_2)$ as label, where v_1 is the first vertex and v_2 is the last vertex vertex in the path induced by $S(v)$. A query 'u reaches v?' is then answered by checking if $C(v_1) \leq C(u) \leq C(v_2)$ when $C(v_2) > C(v_1)$ and by checking if $C(v_1) \leq C(u)$ or $C(v_2) \geq C(u)$ when $C(v_1) > C(v_2)$. □

Combining Lemma 2.4 and Lemma 2.5 we get the labeling we need.

Theorem 2.6. *Let G be planar n-vertex digraph with a set of U interesting vertices given with a plane drawing of G, such that U is covered by the set $\mathcal{F}$ of faces. Then we can, in $O(\min\{n \log n, n|U|\})$ time, compute a reachability labeling of U with labels of size $O(|\mathcal{F}|)$ and constant query time.*

Proof. Suppose the vertex $u \in U$ is in the face F. We use one incoming and one outgoing label of constant size for the vertices in every face in $\mathcal{F} \setminus \{F\}$ from Lemma 2.5, and the constant sized label for the vertices in F from Lemma 2.4. To see that the running time is $O(\min\{n \log n, n|U|\})$, we note that the constructions in Lemma 2.4 and Lemma 2.5 can be done in linear time if we are given a reachability oracle for U with constant query time. To get such an oracle, we can either use Thorup's algorithm or compute the transitive closure of $G[U]$. □

Theorem 2.3 then follows by concatenating the label from the modified recursion and the label we get from Theorem 2.6 for each vertex.

2.3 Concluding remarks

While have an upper bound of $O(\log |\mathcal{F}|)$ on the size of the labels, no non-trivial lower bound on the size of the labels needed for a constant query time reachability labeling scheme is known. This is also true for reachability oracles that do not distribute. It is not clear that allowing more time to answer queries makes smaller reachability oracles possible; it is an open question if there is a reachability oracle for every n-vertex planar digraph using $o(n \log |\mathcal{F}|)$ space even if polylogarithmic query time is allowed.

Note that the upper bound is weakest for the case where the interesting vertices are spread out in many faces. Is there perhaps any property for such maps with many faces that can be used to create smaller labels?

Chapter 3

Reachability substitutes

3.1 Introduction

In this chapter, we consider the problem of constructing small reachability substitutes. Reachability substitutes are defined as follows: Let G be a digraph with $n = |V(G)|$ vertices and let $U \subseteq V(G)$ be a set of κ vertices in G which are designated *interesting*. A *reachability substitute* for (G, U) is a digraph $H = (V', E')$ such that $U \subseteq V'$ and for any pair of vertices $u, v \in U$, there is a path from u to v in H iff there is a path from u to v in G. Throughout the rest of this chapter, κ will denote the cardinality of U. When $U = V(G)$, we also say that H is a reachability substitute of the graph G.

The problem we are interested in is that of finding a *small* reachability substitute, i.e., one that minimizes $|V'| + |E'|$. We show that it is NP-hard. Aside from the complexity of finding a small substitute, we are mainly interested in the structural question of lower and upper bounds on its size and in efficient methods of constructing small substitutes. Trivially, every input has a solution of size $O(\kappa^2)$, and we show by a counting argument that in general it is not possible to obtain a considerably smaller reachability substitute:

Theorem 3.1. *The fraction of pairs (G, U) of digraphs $G = (V, E)$ and subsets $U \subseteq V$ of κ interesting vertices with reachability substitutes of size $o(\kappa^2/\log \kappa)$ can be at most $o(1)$.*

Recall that a Steiner graph $G' = (V', E')$ of a digraph $G = (V, E)$ is an edge-weighted digraph such that $V \subseteq V'$ and the distances d_G and $d_{G'}$ satisfy $d_{G'}(u, v) \geq d_G(u, v)$ for all vertices $u, v \in V$. A Steiner d-preserver is a Steiner graph G' such that $d_{G'}(u, v) = d_G(u, v)$ if $d_G(u, v) \geq d$ [12]. A Steiner 1-preserver of G hence preserves the existence of paths between vertices in $V(G)$, so a Steiner 1-preserver of G is also a reachability substitute. Bollobás, Coppersmith and Elkin proved that every graph on n vertices has a Steiner 1-preserver of size $O(n^2/\log n)$, so each pair (G, U) has a reachability substitute of size $O(|U|^2/\log |U|)$. This shows that the lower bound $O(\kappa^2/\log \kappa)$ on the size of reachability substitutes above is the best possible.

The main focus of this chapter is on planar inputs — the output substitute need not be planar, though. If all vertices of U lie on a constant number of faces of a plane graph, there is a solution of size $O(\kappa \log \kappa)$, which can be found in $O(n \log n)$ time. This result is due to Subramanian [54]. That algorithm was designed as a component of an algorithm for dynamic reachability in planar graphs; the graph is recursively partitioned with small separators, where the separator vertices become the interesting vertices. We will generalize Subramanian's result using Thorup's oracle construction for planar digraphs [57] discussed in Chapter 2. We show that techniques used in a component of Thorup's construction can be adapted to produce a reachability substitute of size $O(\kappa \log \kappa)$ in $O(n \log n)$ time for any planar digraph, regardless of where the interesting vertices are located.

Theorem 3.2. *Any planar n-vertex graph $G = (V, E)$ with a subset $U \subseteq V$ of κ "interesting" vertices has a reachability substitute of size $O(\kappa \log \kappa)$, which can be found in $O(n \log n)$ time.*

Observe that κ may be arbitrarily small compared to the size of the input graph. In view of Theorem 3.1, Theorem 3.2 demonstrates that the structure of reachability relations in planar digraphs is considerably less complex than in general digraphs. Furthermore, the combination of the two theorems implies that most graphs do not have planar reachability substitutes, not of any size.

This observation immediately raises the question how one could characterize the class of directed graphs that have planar substitutes. We make some first observations about such graphs, but we are far from solving it. The full classification problem seems to be a difficult task. We shall address this subject and the questions that arise from it at the end of the chapter.

Road map. The rest of this chapter is organized as follows. In Section 3.2 we derive several complexity results about the computational cost of finding small reachability substitutes and lower bounds on their sizes. In Section 3.3 we sketch a modification of a component of Thorup's oracle construction which yields an $O(\kappa \log \kappa)$-size reachability substitute for any planar graph in $O(n \log n)$ time. Finally, in Sections 3.4 and 3.5 we discuss the implications of our results and point out interesting questions that arise from our work.

3.2 The complexity of reachability substitutes

We begin with an examination of the computational and structural complexity of reachability substitutes.

3.2.1 NP-hardness

Finding a minimum reachability substitute is NP-hard. Our proof is a reduction from the following problem.

Minimum Hitting Set.
Input: A collection $\mathcal{C}$ of subsets of a finite set S.
Output: A *hitting set* for $\mathcal{C}$, i.e., a subset S' of S such that S' contains at least one element from each subset in $\mathcal{C}$.
Objective: Minimize $|S'|$.

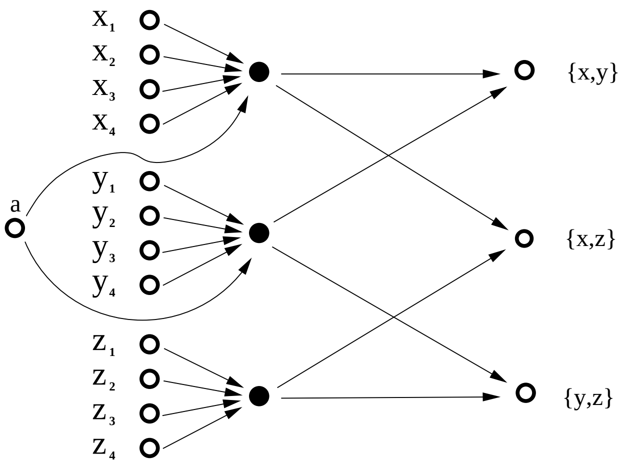

Figure 3.1: A minimum substitute for the graph corresponding to the Minimum Hitting Set instance $\{\{x, y\}, \{x, z\}, \{y, z\}\}$.

We will show that if there is a polynomial-time algorithm that finds a minimum-size reachability substitute, then it can be used to solve the Minimum Hitting Set problem in polynomial time. We will restrict our attention to Minimum Hitting Set instances where each element belongs to at least two sets (this does not make the problem tractable).

The basic idea for our reduction is to construct a graph G with an interesting vertex for every element $x \in S$ and one for every set $C \in \mathcal{C}$, and to demand paths from S to $\mathcal{C}$ to reflect set membership, i.e., we include the edge (x, C) whenever $x \in C$. Additionally, we will have an interesting "start vertex" a from which each set-vertex C must be reachable. A compact substitute for such a graph G should then be forced to establish reachability from this vertex a to the sets in $\mathcal{C}$ by connecting a to the S-vertices and these connections should implicitly encode a small hitting set. There is just one technical problem with this approach: The graph G we are going to construct must already determine which of the elements in S should be reachable from a, because all these vertices are interesting. The trick to circumvent this is to force the reachability substitute to contain uninteresting dummy vertices for the elements of S to which a will then connect. The precise reduction is as follows.

The graph G has a vertex C for each $C \in \mathcal{C}$, one vertex a, and four vertices, x_1, x_2, x_3, x_4, for each element $x \in S$. All of these are interesting and the edges are (a, C) for each $C \in \mathcal{C}$ and (x_i, C) whenever $x \in C$.

Lemma 3.3. *A minimum-size reachability substitute for the graph G above contains an uninteresting vertex $\bar{x}$ for every $x \in S$, such that the interesting vertices reachable from $\bar{x}$ are exactly those sets $C \in \mathcal{C}$ with $x \in C$.*

Proof. Assume otherwise. Let H be a minimum reachability substitute for G and let $x \in S$ be an element such that there does not exist an uninteresting vertex in H that reaches all sets to which x belongs. Each of x_1, x_2, x_3, x_4 reaches exactly those sets that contain x, so we get that each of them must have out-degree at least 2 (recall that we assume that each element belongs to at least two sets). We can assume without loss of generality that x_1, x_2, x_3, x_4 have the same out-neighbors; otherwise, transform the substitute H by connecting all four copies of x to the out-neighbors of the one among them that has minimum out-degree. The subgraph induced by x_1, x_2, x_3, x_4 and their common out-neighbors is a $4 \times k$ bipartite clique for some $k \geq 2$. This contradicts the minimality of H because it can be reduced by adding a non-interesting vertex $\tilde{x}$ and connecting the two sides of this bipartite clique trough $\tilde{x}$. □

We will now show that once we have found a reachability substitute H for G, we can use it to find a minimum hitting set. First, observe that an optimal reachability substitute for G is acyclic. This follows from the fact that G itself contains no directed cycles so that any strongly connected component of a reachability substitute of G would contain at most one interesting vertex. Consequently, we could collapse any such component to just one vertex, thereby reducing the size of the substitute.

We now use the acyclicity to normalize a reachability substitute for G: Consider an optimal reachability substitute H for G. For each edge (a, b) of H, we move the tail b of this edge to some non-interesting predecessor of b that has only interesting in-neighbors.

Lemma 3.4. *After applying the normalizing step above to an optimal reachability substitute for G, all of the out-neighbors of a are from among the $\tilde{x}$ vertices whose existence is guaranteed by Lemma 3.3.*

Proof. Since the reachability substitute is acyclic, it is clearly not optimal if it has uninteresting source vertices; they can just be removed. By Lemma 3.3, for every $x \in S$ the reachability substitute has an uninteresting vertex $\tilde{x}$ that reaches exactly the sets that x belongs to. The substitute is not optimal if $\tilde{x}$ is not unique or if x_1, x_2, x_3, x_4 have out-neighbors other than $\tilde{x}$. The lemma follows. □

Lemma 3.4 implies that the out-neighbors of a correspond to a hitting set for $\mathcal{C}$. In the optimal substitute, this hitting set must be of minimum cardinality. Hence:

Theorem 3.5. *Finding a minimum reachability substitute for a given digraph is NP-hard.*

3.2.2 Incompressibility of almost all digraphs

We now prove Theorem 3.1. A counting argument will show that there are relations that cannot be represented by a graph of size less than $\Omega(\kappa^2/\log\kappa)$. In fact, it turns out that almost all digraphs allow for almost no compression, which is quite in contrast to the planar case of Theorem 3.2.

Let $\kappa = 2k$ for some integer k and consider as possible inputs all labeled bipartite graphs with k (interesting) vertices in each bipartition (the labels are from $\{1, 2, \ldots, \kappa\}$), with all edges going in the same direction. There are exactly

2^{k^2} such input graphs and no two of them induce the same reachability relation. We use this as a lower bound on the total number of inputs.

On the other hand, we upper bound the number $N(\ell)$ of different reachability substitutes of size at most ℓ. Obviously, $N(\ell)$ is smaller than the number of different digraphs on ℓ vertices with at most ℓ edges. The latter quantity can be bounded as follows: Fix a labeling of the vertices. For any edge, there are less than ℓ^2 possibilities of how to place it in (or omit it from) the digraph. As a consequence, the number of digraphs, and thus $N(\ell)$, is bounded by $\ell^{2\ell} = 2^{2\ell \log \ell}$. Therefore, only a fraction of $2^{2\ell \log \ell - k^2}$ of all inputs can have a reachability substitute of size at most ℓ. This fraction can only be constant if $\ell \in \Omega(k^2 / \log k)$.

3.2.3 A lower bound for planar outputs

The almost linear-size reachability substitutes for planar graphs that we construct for Theorem 3.2 will, however, be far from planar. To see that this cannot be avoided, we argue why in general, planarity must be sacrificed if one wants small reachability substitutes.

Theorem 3.6. *The planar digraph in Figure 3.2 with κ interesting vertices has no planar reachability substitute of size $o(\kappa^2)$.*

Proof. Consider the plane digraph G in Figure 3.2. The paths through the black uninteresting vertices are set up to make a lower interesting vertex v_j reachable from some upper interesting vertex u_i if and only if $i \le j < i + r$.

We claim that any planar reachability substitute for G' with the white vertices marked interesting, must contain essentially all those black intermediate vertices, too. In other words, G is incompressible if planarity is to be maintained. With $r \approx \ell$ this gives a quadratic lower bound on the representation size in terms of κ.

To prove this claim, we first observe that a path from s to a vertex u_i can only intersect paths from s to other vertices u_j in the upper row. If a path from s to u_i intersects a path from any other vertex than s or u_i, it will induce a new path that is not allowed in the substitute. On the other hand, if a path from u_i intersects the path from s to u_i, we get a cycle with only one interesting vertex. We can then contract the cycle into a single vertex and get a smaller substitute.

The same observation is true for the paths from vertices v_j to t. Hence, if we fix a plane drawing of a minimum planar reachability substitute H of G, we can add one Jordan curve J_u through all the u_is and another Jordan curve J_v through all the v_js such that all the paths between the u_is and the v_js must be outside these curves. That is, if we remove the paths from s and to t which do not contain any other interesting vertices, there will be one face in which all the u_is are contained and one face in which all the v_js are contained.

Now, consider a vertex u_i. The Jordan curve J_v together with the paths from u_i to v_i and v_{i+r-1} delimits a region R in the plane. By planarity, the only interesting vertices that can be inside R are $\{v_{i+1}, \ldots, v_{i+r-2}\}$.

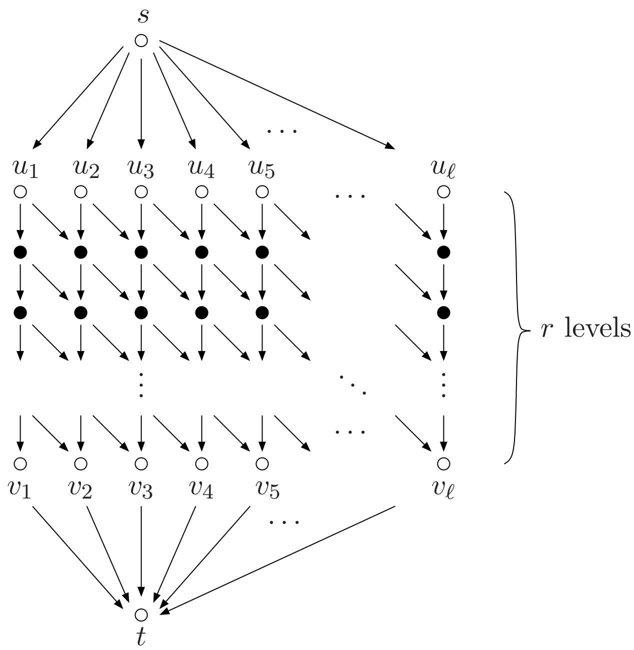

Figure 3.2: A planar acyclic digraph with no planar substitute of sub-quadratic size.

Claim A. All the vertices in $\{v_{i+1}, \ldots, v_{i+r-2}\}$ are inside R.
Proof. Suppose not. Then v_{i+k} is outside R for some $0 < k < r - 1$. Either the paths from u_i to v_i and v_{i+k} or the paths from u_i to v_{i+k} and v_{i+r-1} delimits a new region R' containing R together with J_v. W.l.o.g. assume R' is delimited by the paths to v_i and v_{i+k}. Then v_{i+r-1} is inside R'. But there must be a path between the vertex u_{i+r-1} and v_{i+r-1}, and this path is not allowed to cross any of the paths delimiting R'; a contradiction. △

By a similar argument, we see that the $v_1, v_2, \ldots, v_\ell$ must come in this order while traversing J_v either clockwise or counterclockwise.

Now, consider a pair of vertices $u_i, u_{i'}$ with $i \leq i' < i + r$. The path from $u_{i'}$ to $v_{i'}$ is not allowed to cross any path to v_i. Hence, in a substitute for G', there must be a path P from u_i to v_{i+r-1} and a path P' from $u_{i'}$ to $v_{i'}$ that intersect at some vertex x. It is easy to see that the set of v_js reachable from this x is exactly $\{v'_i, \ldots, v_{i+r-1}\}$. Hence, for each such pair of vertices $u_i, u_{i'}$ there must be a different vertex x in the substitute. Setting $r \in O(\ell)$ gives us desired lower bound.

□

3.3 An $O(\kappa \log \kappa)$-size substitute for planar digraphs

Thorup's oracle construction [57] that we sketched in Section 2.1 can be turned into an algorithm that produces a reachability substitute. We start by reducing the problem to 2-layered graphs as before. However, to create a reachability substitute, we don't need to pass down a frame with each recursive call. Hence, the frame reducing recursion is removed. Like before, we find the connections to and from the directed separator paths. To get a reachability substitute, we add these connections as edges. For each vertex u in the graph and each directed separator path P, let P_u be the first vertex along P that is reachable from u and let P^u be the last vertex on P that reaches u. The reachability substitute contains two edges between u and P: one from u to P_u and one from P^u to u.

After the edges to and from the separator paths are added, these paths are contracted into the root of the spanning tree. When a new separator is computed, the root is given weight 0, since all paths through the root already have been added to the substitute. The depth of the recursion remains $O(\log n)$ and the number of edges inserted into the substitute in each recursive level is $O(n)$, so the total size of the reachability substitute is $O(n \log n)$.

We now sketch how this algorithm can be modified to give an $O(\kappa \log \kappa)$-size substitute when only a subset of κ vertices are interesting. Two changes are necessary. The first is to reduce the recursion depth to $O(\log \kappa)$ and the second is to reduce the number of edges inserted into the reachability substitute in each recursive level to $O(\kappa)$. This is very similar to the change we made to the oracle construction in the previous chapter.

To reduce the recursion depth, we change the vertex weighting when used when we compute a separator: the interesting vertices get weight 1 and the other vertices get weight 0. The number of interesting vertices in each connected component after removing the separator is at most a constant fraction of the number of interesting vertices we had before, and the depth of the recursion is thus $O(\log \kappa)$.

Finally, the contribution of each recursive level to the size of the substitute can be reduced to $O(\kappa)$ as follows. Initially, we link each interesting vertex u to P_u and P^u as before. This adds at most $O(\kappa)$ edges to the substitute; two for each interesting vertex and directed separator path, and the number of such paths is constant. The problem now is that the separator paths themselves may consist of too many vertices and edges. In each directed path, there are $O(\kappa)$ vertices to which we directly connected interesting non-separator vertices. We call them "connection points". Any separator vertex that is not a connection point has exactly one incoming edge and one outgoing edge, so we remove it and connect its predecessor directly to its successor.

This way we get the desired $O(\kappa \log \kappa)$-size substitute for Theorem 3.2.

3.4 Planarly induced graphs

As mentioned in the introduction, a consequence of Theorems 3.1 and 3.2 is that most graphs do not have planar reachability substitutes, not of any size. We say that a graph is *planarly induced* if it has a planar reachability substitute. The class of planarly induced graphs is clearly very different from the class of planar graphs. For example, an instance of $K_{3,3}$ in which all edges are oriented from one bipartition to the other, is not planar but is induced by a 7-vertex 3-layered planar graph. On the other hand, the graph obtained by subdividing each edge in the oriented $K_{3,3}$ above, shown in Figure 3.3, is not planarly induced. This can be seen as follows.

In a planar substitute for that graph, there would have to exist a path from any of the three sources to any of the three sinks, which also has to pass through one of the nine intermediate interesting vertices. It can be seen that any crossing between two such paths would induce a path between a source or sink vertex and an intermediate vertex who are not linked in the given graph. Therefore, a planar substitute cannot exist for this graph, by Kuratowski's theorem. Similarly, it follows that there is no planar reachability substitute for such an oriented subdivision of any non-planar bipartite graph.

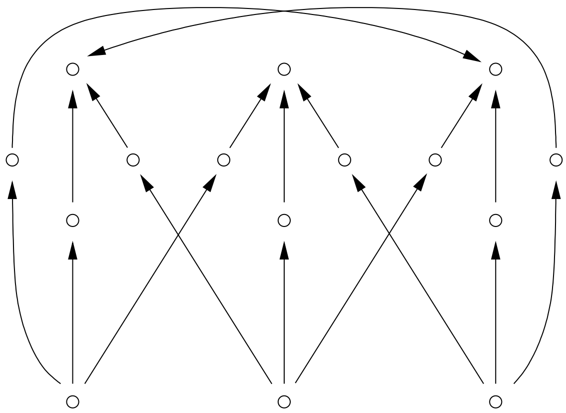

Figure 3.3: A directed $K_{3,3}$-subdivision that is not planarly induced.

Several questions arise from this observation. Are there alternative characterizations of the planarly induced graphs? One might suspect that planarly induced graphs could be captured by a generalization of Kuratowski's classification of planar graphs. However, the example above shows that planarly induced graphs are not even stable under subdivisions, a property that one would naturally expect when trying to follow some graph-minor related approach.

Further open questions concern the minimal size of planar substitutes. Given a planarly induced graph with κ vertices, how large is its smallest planar reachability substitute? We have shown that the answer can be as bad as $\Omega(\kappa^2)$ in the worst case. Is this also an upper bound?

On the algorithmic side, we ask whether planarly induced graphs can be identified by an efficient algorithm. Furthermore, if we know that a graph is planarly induced, what is the complexity of constructing the smallest possible planar reachability substitute for it? Or an approximation of the smallest possible planar substitute? Or any planar substitute?

3.5 Further open problems

Apart from the open problems related to the concept of planarly induced graphs, mentioned above, there are further immediate open questions that arise from the results in this chapter.

There remains a gap between the upper bound of $O(\kappa \log \kappa)$ and the trivial lower bound of $\Omega(\kappa)$ on the size of a reachability substitute for a planar input.

So far, we have not found a planar instance that does not have a linear-size reachability substitute. Even for the special case in which all interesting vertices are on a single face of the graph, it is an open problem to improve upon the $O(\kappa \log \kappa)$-size solution (which, as we have mentioned, was first achieved by Subramanian [54]). It seems like it should be possible to turn the constant size labeling from the previous chapter into a substitute of size $O(\kappa)$ in this case, but so far our efforts in this direction has been fruitless.

We have shown that the problem of computing a minimum-size reachability substitute for a general graph is NP-hard. Can it be approximated? Is it NP-hard also for planar inputs? If so, can it be approximated within a better factor than our $O(\log \kappa)$?

Finally, it would be interesting to find other classes of graphs that have small reachability substitutes.

Chapter 4

The order dimension of planar maps

4.1 Introduction

In this chapter we study planar maps and the order dimension of posets related to them. By a planar map M we mean the combinatorial data given by the set V of vertices, the set E of edges, the set F of faces and the incidence relations between these sets. As usual, the dual map of M is denoted M^*.

Most of the maps we consider in this chapter are outerplanar. We differentiate between two notions of outerplanar maps. A planar map $M = (G, D)$ is *weakly outerplanar* if G is outerplanar, and *strongly outerplanar* if G is outerplanar all the vertices are on the boundary of the outer face in D. When it is clear from the context, the qualifiers weakly and strongly will be omitted.

The dimension is a widely studied parameter of posets. Since its introduction by Dushnik and Miller [24] in 1941, dimension has moved into the core of combinatorics. There are close connections and analogies with the chromatic number of graphs and hypergraphs. From the applications point of view, dimension is attractive because low dimension warrants a compact representation of the poset. Trotter [58] provides an extensive introduction to the area.

The vertex-edge-face poset $\mathbf{P}_M$ of a planar map M is the poset on the vertices, edges and faces of M ordered by inclusion. The vertex-face poset $\mathbf{Q}_M$ of M is the subposet of $\mathbf{P}_M$ induced by the vertices and faces of M.

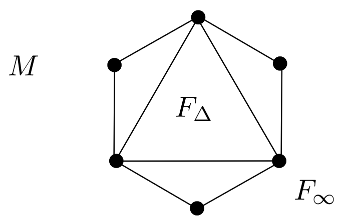

Figure 4.1: A planar map. Two faces are labeled, F_∞ is the *outer face*.

Note that if M is connected, the vertex-edge-face poset $\mathbf{P}_{M^*}$ of the dual map, is just the dual poset $(\mathbf{P}_M)^*$ (i.e., $x < y$ in $(\mathbf{P}_M)^*$ if and only if $y < x$ in $\mathbf{P}_M$). The same observation is true for $\mathbf{Q}_M$.

Recall that the *dimension* $\dim(\mathbf{P})$ of a poset $\mathbf{P}$ is the minimum number t

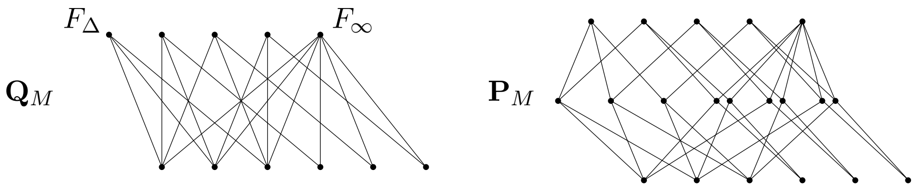

Figure 4.2: The vertex-face and the vertex-edge-face posets of the map from Figure 4.1.

such that $\mathbf{P}$ is the intersection of t linear orders on the same ground set. Our investigations ground on Schnyder's characterization of planar graphs in terms of dimension [53] and the two theorems of Brightwell and Trotter cited below. A simpler proof of Theorem 4.2 can be found in [32].

Theorem 4.1 (Brightwell and Trotter [15])**.** *If M is a planar map, then* $\dim(\mathbf{P}_M) \leq 4$.

Theorem 4.2 (Brightwell and Trotter [14])**.** *If M is a 3-connected planar, then* $\dim(\mathbf{Q}_M) = 4$.

The cases where the dimension is 2 are well-studied. There are fast algorithms to test whether a poset $\mathbf{P}$ is of dimension 2, see e.g. [46]. Moreover, Brightwell and Trotter [15] have presented a complete characterization of the planar maps with $\dim(\mathbf{P}_M) = 2$. This characterization can be turned into a linear time recognition algorithm. Surprisingly, the characterization of $\dim(\mathbf{Q}_M) = 2$ is less satisfying. It seems that the most compact way of stating the characterization is to refer to the list of 3-irreducible posets, see e.g., [58, Table 2]. A map that contains $\mathbf{A}_k$ for some $k \geq 4$ in its vertex-face poset also contains another forbidden subposet. Therefore, it is enough to forbid $\mathbf{A}_3$, $\mathbf{B}$, $\mathbf{CX}_2$ and $\mathbf{EX}_1$ and their duals, i.e, subposets with six or seven elements (see Figure 4.3). Another option is to refer to characterizations of bipartite permutation graphs, e.g, as being exactly the bipartite AT-free graphs, see [13] for more on this topic. Again, these characterizations lead to linear time recognition algorithms, see [13, 16].

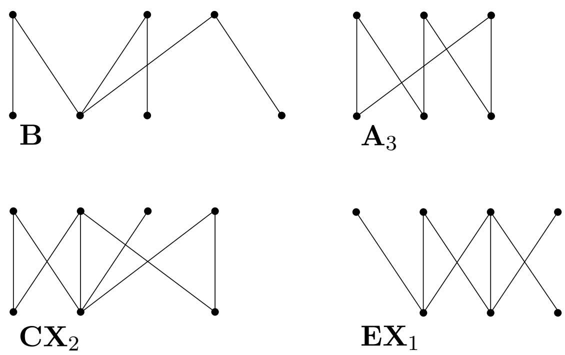

Figure 4.3: The forbidden posets $\mathbf{B}$, $\mathbf{A}_3$, $\mathbf{CX}_2$ and $\mathbf{EX}_1$.

For $\dim(\mathbf{P}_M) \geq 3$ it is sufficient that M has a vertex of degree 3. To test for dimension 3 is known to be NP-complete [60]. A major open problem in the area is to determine the complexity of the dimension 3 problem for orders of height 2. Even the special case where the order of height 2 is the vertex-face poset of a planar map remains open. Motivated by these algorithmic questions we approach the problems of characterizing the maps M with $\dim(\mathbf{P}_M) \leq 3$ and the maps with $\dim(\mathbf{Q}_M) \leq 3$. These characterization problems have earlier been posed by Brightwell and Trotter [15].

4.1.1 Our contributions

In Section 4.2 we prove that for $\dim(\mathbf{Q}_M) \leq 3$ it is necessary that M is K_4-subdivision free. For $\dim(\mathbf{P}_M) \leq 3$ an additional necessary condition is that both M and M^* are $K_{2,3}$-subdivision free. This means that if $\dim(\mathbf{P}_M) \leq 3$, then both M and M^* are outerplanar.

In Section 4.3, we study the simplest class of maps M such that M and M^* are outerplanar. We call these maps *path-like*. For maximal path-like maps we prove that $\dim(\mathbf{P}_M) \leq 3$ is equivalent to the existence of a special oriented coloring of the interior edges and characterize the path-like maps which admit such a coloring. The characterization is turned into a linear time algorithm that generates a 3-realizer, i.e., three linear extensions whose intersection is $\mathbf{P}_M$, or returns the information that $\dim(\mathbf{P}_M) \geq 4$.

Finally, in Section 4.4, we prove that if M is 2-connected and M and M^* are outerplanar, then $\dim(\mathbf{Q}_M) \leq 3$. We also present a strongly outerplanar map with a vertex-face poset of dimension 4. The example, a maximal outerplanar graph with 21 vertices, is quite large. We provide some arguments which indicate that our example is not far from being as small as possible.

4.2 Vertex-edge-face posets of dimension at most 3

From Theorem 4.2, we know that $\dim(\mathbf{Q}_M) = 4$ for every 3-connected map M. We show that this excludes K_4-subdivisions from being contained in M if $\dim(\mathbf{Q}_M) \leq 3$.

Theorem 4.3. *Let M be a planar map that contains a subdivision of K_4. Then* $\dim(\mathbf{Q}_M) > 3$.

Proof. We will prove that $\mathbf{Q}_M$ has the vertex-face poset of some 3-connected planar map as a subposet, and then apply the Brightwell-Trotter Theorem. This is essentially done in two steps: first 1-vertex cuts and then 2-vertex cuts are removed.

A K_4-subdivision in M will be contained in a 2-connected component of M. The vertex-face poset of a 2-connected component of M is an induced subposet of $\mathbf{Q}_M$. Hence, we can assume that M is 2-connected.

Now, consider a 2-vertex cut $\{x, y\}$. There must be two components C_1 and C_2 such that removing x and y separates C_1 from C_2. We create two new maps by replacing one of the two components by an edge $\{x, y\}$, see Figure 4.4. The vertex-face posets of these new maps are subposets of $\mathbf{Q}_M$. Furthermore, if M contains a K_4-subdivision, one of the new maps must contain a K_4-subdivision. □

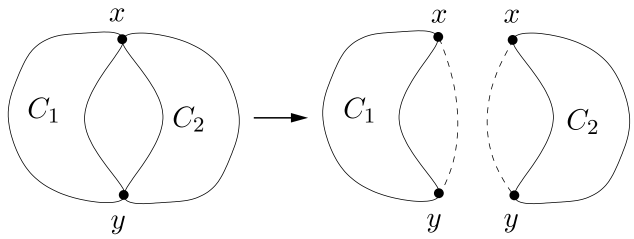

Figure 4.4: Removing a separating pair.

For vertex-edge-face posets, we provide another criterion which forces dimension 4. If M contains a $K_{2,3}$-subdivision, then $\dim(\mathbf{P}_M) = 4$. Before analyzing the general situation we deal with the simple case where M is, actually, a subdivision of $K_{2,3}$.

Proposition 4.4. *Let M be a planar drawing of a subdivision of $K_{2,3}$. Then* $\dim(\mathbf{P}_M) > 3$.

Proof. Let x and y be the two vertices of degree 3 in M, and let P_1, P_2 and P_3 be the three x-y paths. The map has three faces F_1, F_2 and F_3. In Figure 4.5 face F_i is labeled R_i. The vertex closest to y in the path P_i is denoted v_i.

Suppose $\{L_1, L_2, L_3\}$ is a realizer of $\mathbf{P}_M$. By symmetry, we may assume that $y > x$ in L_1 and L_2 and $x > y$ in L_3. The edge $\{v_1, y\}$ can go below x only in L_3. In L_3 we thus have v_1 below $\{v_1, y\}$ below x below F_1, F_2 and F_3. In the same way, we obtain that v_2 and v_3 are below F_1, F_2 and F_3 in L_3. Hence, none of the three critical pairs of the the subposet $\mathbf{S}$ induced by v_1, v_2, v_3, F_1, F_2 and F_3 in $\mathbf{P}_M$ can be reversed in L_3. However, $\mathbf{S}$ is a crown with $\dim(\mathbf{S}) = 3$. This shows that $\{L_1, L_2, L_3\}$ is not a realizer of $\mathbf{P}_M$. Hence, $\dim(\mathbf{P}_M) > 3$. □

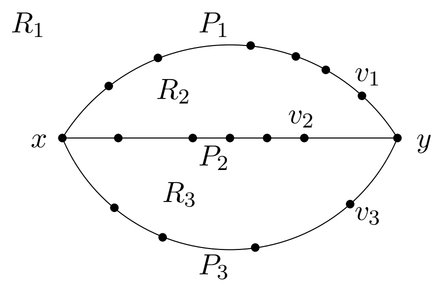

Figure 4.5: The three paths P_1, P_2 and P_3 partitions the map into 3 regions.

For the general case, where M only contains a $K_{2,3}$-subdivision, we have to use a more sophisticated technique. We illustrate this technique with an alternative proof of the simple case.

Second proof of Proposition 4.4. The vertices and edges of path P_i all belong to $F_i \cap F_{i+1}$ (cyclically). Hence, each path P_i induces a fence of the form $x < e_0 > u_1 < e_1 > \ldots u_s < e_s > y$ between x and y in $\mathbf{P}_M$ such that all maximal elements are below F_i and F_{i+1}. These three fences are mutually disjoint.

Suppose $\{L_1, L_2, L_3\}$ is a realizer of $\mathbf{P}_M$. By symmetry, we may assume that $y > x$ in L_1 and L_2 and $x > y$ in L_3. Now, consider the fence induced by P_i, $i \in 1, 2, 3$, see Figure 4.6.

The edge $\{v_i, y\}$ must be below x in L_3, hence v_i is below x in L_3. Let w_i be the last vertex encountered when traversing the path P_i from y to x which is below x in L_3, and let e_i be the edge leaving w_i in direction of x. The choice of w_i implies that e_i is above x and y in L_3. Since e_i has to go below y somewhere there is an index $j_i \in \{1, 2\}$ such that e_i and thus w_i go below y in L_{j_i}.

Two of the three indices j_1, j_2, j_3 must be equal, so we can w.l.o.g. assume that w_1 and w_2 are below all faces that contain both x and y in in L_2 and L_3.

Now, none of the critical pairs of the subposet **2+2** of $\mathbf{P}_M$ induced by w_1, w_2, F_1 and F_3 are reversed in L_2 or L_3. But $\dim(\mathbf{2+2}) = 2$, so the critical pairs of $\mathbf{Q}$ cannot be reversed in L_1 alone. Hence $\{L_1, L_2, L_3\}$ cannot be a realizer of $\mathbf{P}_M$. □

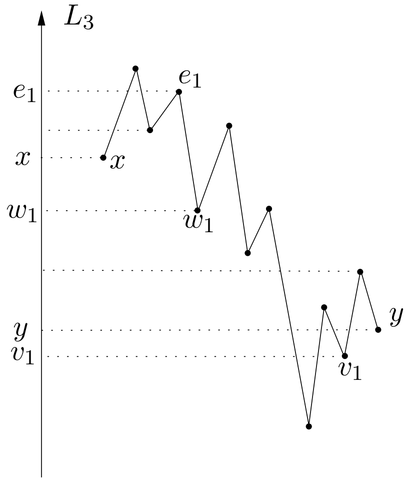

Figure 4.6: The fence of the path P_1 and L_3.

We now move on to the slightly more complicated case where M only contains a subdivision of $K_{2,3}$.

Theorem 4.5. *Let M be a planar map such that M contains a subdivision of $K_{2,3}$. Then* $\dim(\mathbf{P}_M) > 3$.

Proof. If M contains a subdivision of K_4, the conclusion of the lemma follows from Theorem 4.3. We thus can assume that M contains no subdivision of K_4.

Let x and y be the degree 3 vertices in the subdivision of $K_{2,3}$. Our goal is to find at least three mutually disjoint fences $\mathbf{T}_i$ between x and y, and a set of faces F_i such that $x, y \in F_i$ and each minimal element in $\mathbf{T}_i$ is below F_i and F_{i+1}.

Given fences $\mathbf{T}_i$ and faces F_i as described we can continue as in the previous proof: Assume a realizer $\{L_1, L_2, L_3\}$ such that $y > x$ in L_1 and L_2 and $x > y$ in L_3. In each fence $\mathbf{T}_i$ we find a minimal element w_i which is below x in L_3 and below y in some L_{j_i}, $j_i \in \{1, 2\}$. Since $i \geq 3$ there are indices a and b with $j_a = j_b$, and we can w.l.o.g. let $j_a = j_b = 1$. Let $a' \in \{a, a + 1\}$ and

$b' \in \{b, b+1\}$ be such that $w_b \notin F_{a'}$ and $w_a \notin F_{b'}$. Hence, $w_a, F_{a'}, w_b, F_{b'}$ induce a **2+2**. The critical pairs of this **2+2** are not reversed in L_3 nor in L_1, and they can't be both be reversed in L_2. This is in contradiction to the assumption that $\{L_1, L_2, L_3\}$ is a realizer. Hence, $\dim(\mathbf{P}_M) > 3$.

It remains to show how to determine appropriate fences $\mathbf{T}_i$. Consider a maximal set $P_0, P_1, \ldots, P_k$ of pairwise internally disjoint paths from x to y. Clearly, $k \geq 2$. The numbering should correspond to the cyclic order of their first edges at x. Let R_i be the bounded area between P_{i-1} and P_i. The maximality of the family $P_0, P_1, \ldots, P_k$ implies that in R_i there is a face F_i that has a nonempty intersection with the interior of both, P_{i-1} and P_i. Next we prove that this face F_i contains x and y.

Claim A. In R_i there is a face F_i containing x and y.

Otherwise, the cycle consisting of P_i and P_{i+1} has a chordal path and this path, together with P_i, P_{i+1} and some P_j, $j \notin \{i, i+1\}$ is a subdivision of K_4 in M. △

Let u and w be vertices of P_i such that u is closer to x than w and $(u, w) \neq (x, y)$. A *shortcut* between u and w over P_i is a path from u to w which is internally disjoint from P_i. Two shortcuts, $\{u_1, w_1\}$ and $\{u_2, w_2\}$ are *crossing* if their order along P_1 is either u_1, u_2, w_1, w_2 or u_2, u_1, w_2, u_1. In particular this requires the four vertices to be pairwise different.

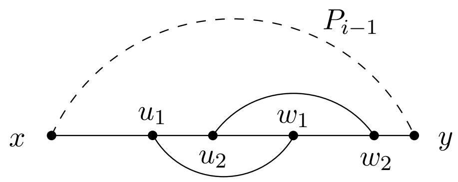

Figure 4.7: Crossing shortcuts.

Claim B. There is no crossing pair of shortcuts on P_i.

Otherwise, the four vertices of the two shortcuts are the degree three vertices of a subdivision of K_4. This subdivision of K_4 is formed by the shortcuts together with P_i and P_{i-1}. See Figure 4.7. △

Let V_i be the set of all vertices of P_i that are contained in $F_i \cap F_{i+1}$.

Claim C. Two consecutive vertices u and w in V_i either are the two endpoints of an edge or there exists a face F such that $F \cap V_i = \{u, w\}$.

Suppose $\{u, w\}$ is not an edge. From Claim B it follows that there is a shortcut $\{u, w\}$ over P_i. Essentially the same proof as for Claim A shows that the subregion bounded by P_i and the shortcut between u and w contains a face F with $u, w \in F$; otherwise, there is a K_4-subdivision. △.

The fence $\mathbf{T}_i$ consists of V_i (the set of minimal elements) and edges, respectively faces over consecutive pairs of vertices in V_i. The existence of a $K_{2,3}$-subdivision between x and y implies that at least three of the fences $\mathbf{T}_0, \mathbf{T}_1, \ldots, \mathbf{T}_k$ are nontrivial, i.e., have minima different from x and y. These fences can be used to conclude the proof. □

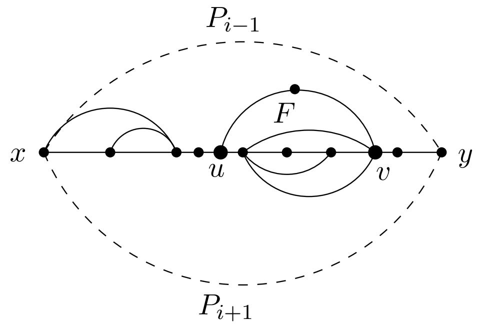

Figure 4.8: The common face F of u and w.

Theorem 4.6. *If* $\dim(\mathbf{P}_M) \leq 3$*, then* M *and* M^* *are both weakly outerplanar.*

Proof. From Theorem 4.3 and Theorem 4.5, we know that if $\dim(\mathbf{P}_M) \leq 3$ then M contains neither a K_4-subdivision nor a $K_{2,3}$-subdivision. This is equivalent to saying that the graph G corresponding to the map M is outerplanar. Since $\mathbf{P}_M$ and $\mathbf{P}_{M^*}$ are dual orders and, hence, have the same dimension the same necessary condition for $\dim(\mathbf{P}_M) \leq 3$ applies to M^*. □

Note that testing if M and M^* are weakly outerplanar, i.e., if the corresponding graphs are outerplanar, can be done in linear time [45].

4.3 Path-like maps and permissible colorings

From Theorem 4.6 we know that if $\dim(\mathbf{P}_M) \leq 3$, both M and M^* are weakly outerplanar. In this section we study the order dimension of 2-connected maps M, such that M is strongly outerplanar and M^* is weakly outerplanar.

A 2-connected component of an outerplanar map M has a Hamilton cycle. If the graph of M is simple, the Hamilton cycle is unique. This yields a natural partition of the edges of M into *cycle edges* and *chordal edges.* The restriction of the dual graph to the graph induced by the vertices corresponding to bounded faces is called the *interior dual.* For a strongly outerplanar map, the edges of the interior dual are just the dual edges of the chordal edges.

We say that a simple 2-connected outerplanar map M is *path-like* if and only if the interior dual of M is a simple path. Note that this implies that the Hamilton cycle is the boundary of the outer face F_∞, i.e. that M is strongly outerplanar. Since the interior dual is a path, it follows that M^* is weakly outerplanar. On the other hand, if M is a 2-connected strongly outerplanar map and M^* is weakly outerplanar, the interior dual of M must be a simple path. Hence, M is path-like iff M is a 2-connected outerplanar map such that M^* is weakly outerplanar.

Path-like maps are in some sense the simplest ones with $\dim(\mathbf{P}_M) \leq 3$. From Theorem 4.6 it follows that if M is a 2-connected strongly outerplanar map with $\dim(\mathbf{P}_M) \leq 3$, M must be path-like. We can also prove something slightly stronger.

Proposition 4.7. *Let M be a simple 2-connected planar map with* $\dim(\mathbf{P}_M) \leq 3$*. The map M' obtained by moving all the chordal edges of M into the interior of the Hamilton cycle is path-like.*

Proof. Suppose not. Then the interior dual of M' contains a vertex of degree at least 3, and hence its dual $(M')^*$ contains a subdivision of $K_{2,3}$ with the dual of one degree 3 vertex inside the Hamilton cycle H and the dual of the other outside. We proceed to show that we can move the necessary chordal edges outside one by one to create M without destroying the $K_{2,3}$-subdivision in the dual.

We do this as follows: let $M' = M_0, M_1, \ldots, M_k = M$ be a sequence of maps such that M_{j+1} is obtained from M_j by moving a chordal edge from the inside to the outside of the Hamilton cycle. The proposition follows from the following claim.

Claim A. For each map M_i, $i = 0, 1, \ldots, k$, the dual map M_i^* contains a $K_{2,3}$-subdivision such that H^* separates the two vertices of degree 3.

We prove the claim by induction on i. We have already seen that the statement is true for $M_0 = M'$.

Suppose the claim is true for M_i. Let $e = \{u, v\}$ be the edge that has to be moved to the outside of H to get from M_i to M_{i+1}. Let F^* and G^* be the degree 3 vertices in the $K_{2,3}$-subdivision in M_i^*, where F is inside H. We construct a new map M_i', by adding the edge $e' = \{u, v\}$ to M_i outside H, see Figure 4.9. Note that F and G must be on the same side of the cycle $\{e, e'\}$, since otherwise $\{e^*, (e')^*\}$ is a 2-edge cut in $(M_i')^*$ separating F^* from G^*; a contradiction.

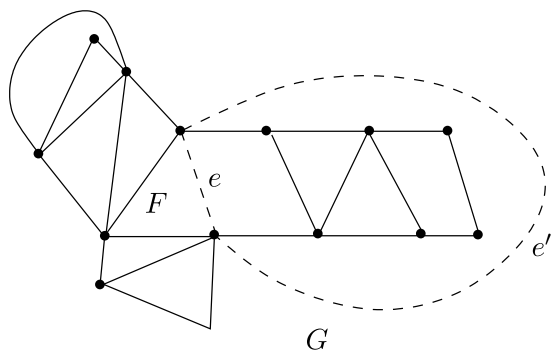

Figure 4.9: The map M_i' is constructed by adding e' to M_i.

Claim B. Let P^* be a simple F^*-G^* path in $(M_i')^*$. If $e^* \in P^*$, then P^* consists of at least 3 edges.

Suppose $e^* \in P^*$. Since P^* is simple, and F and G are on the same side of the cycle $\{e, e'\}$, it follows that $(e')^* \in P^*$. Moreover, the dual H^* of the Hamilton cycle H separates F^* from G^*, so P^* must contain the dual of a cycle edge. But neither e nor e' are cycle edges, so the claim follows. △

Now, M_{i+1} is obtained by removing e from M_i'. In $(M_i')^*$, this corresponds to the contraction of edge e^*. If e^* is not on any F^*-G^* path, then M_{i+1}^* contains a $K_{2,3}$-subdivision. On the other hand, an F^*-G^* path containing e^* has at least 3 edges, hence, the contraction of e^* cannot destroy the $K_{2,3}$-subdivision. □

Corollary 4.8. *Let M' be obtained from a simple weakly outerplanar map M by flipping all chordal edges to the interior of the Hamilton cycle. If $(M')^*$ contains a $K_{2,3}$-subdivision, then so does M^*.*

In the rest of this section, we consider *maximal path-like maps*, i.e., path-like maps where all interior faces are triangles. Consider a triangle of a maximal path-like map M. Each of the three vertices forms a critical pair with a face or edge that is above the other two vertices of the triangle. In Figure 4.10 these critical pairs are (u, F_u), (v, F_v), (w, e_w). Note that one of F_u or F_v can be an edge, if the triangle has two edges in the Hamilton cycle. Now, if $\dim(\mathbf{P}_M) = 3$, we can color each of these critical pairs with the linear extension it is reversed in to obtain a 3-coloring of the angles. For convenience we interchangeably use red, green and blue or 1,2 and 3 as the of colors.

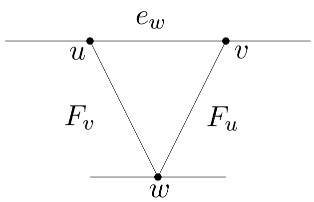

Figure 4.10: The critical pairs of a triangle.

We go on to prove some properties of such an angle 3-coloring for a maximal path-like map M with $\dim(\mathbf{P}_M) \leq 3$.

Lemma 4.9. *No two angles in a triangle can have the same color.*

Proof. Consider a triangle with the angle coloring described above. Any two of the critical pairs (u, F_u), (v, F_v), (w, e_w) form an alternating cycle. Hence, no two pairs can be reversed in the same linear extension. □

Lemma 4.10. *Let $e = \{a, b\}$ be a chordal edge. The four angles α_ℓ, α_r, β_ℓ and β_r incident on e at a and b are colored such that all three colors are used, and one of the pairs (α_ℓ, α_r) or (β_ℓ, β_r) is monochromatic.*

Proof. We refer to Figure 4.11. Suppose $\alpha_\ell = 1$ and $\alpha_r = 2$. This implies that in L_1 we have F_b^ℓ and F_b^r above a above F_a^ℓ above b. In L_2 we have the same order with F_a^r taking the role of F_a^ℓ. Hence b has to be above both F_b^ℓ and F_b^r in L_3 which is equivalent to $\beta_\ell = \beta_r = 3$.

Suppose both pairs of angles have the same colors, say $\alpha_\ell = \alpha_r = 1$ and $\beta_\ell = \beta_r = 2$ Then the third angle in both triangles (at x and y, respectively) must have color 3. This induces a monochromatic alternating cycle (x, F_x), (y, F_y), a contradiction. □

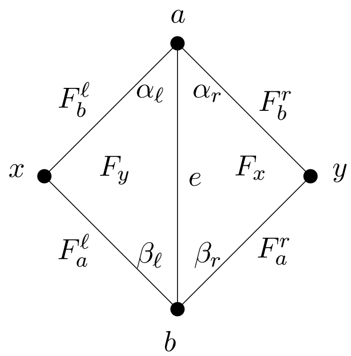

Figure 4.11: Colors and critical pairs around a chordal edge.

By Lemma 4.10 we can encode the angle coloring as an oriented coloring of the chordal edges: each chordal edge gets the color that appears twice around it and is oriented towards the endpoint where this happens.

The *orientation* of an interior triangle is is either *clockwise* or *counterclockwise* depending on the cyclic reading which shows the colors 1,2,3 in this order. Lemma 4.9 implies that the orientation of interior triangles is defined.

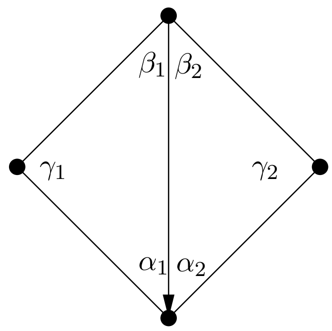

Figure 4.12: The triangles must have the same orientation.

Lemma 4.11. *All interior triangles have the same orientation.*

Proof. This is a direct consequence of Lemma 4.10 and Lemma 4.9. Referring to Figure 4.12 we discuss one of the cases: Suppose the left triangle in the figure is counterclockwise, i,e., $(\alpha_1, \beta_1, \gamma_1) = (i, i+1, i+2)$. The orientation of the edge implies $\alpha_2 = i$, hence from Lemma 4.10 we get $\beta_2 = i + 2$. This shows that the right triangle is counterclockwise as well. □

Lemma 4.12. *Let c be the color of the chordal edge e. Then $e > F_\infty$ in L_c.*

Proof. Again referring to Figure 4.12 we observe that $\gamma_1 \neq \gamma_2$ and $c \neq \gamma_1, \gamma_2$ by Lemma 4.11. Hence, $e < F_\infty$ in L_{γ_1} and L_{γ_2}. Therefore $e > F_\infty$ in L_c. □

Lemma 4.13. *A vertex is either a sink or a source w.r.t. the orientation of the chordal edges.*

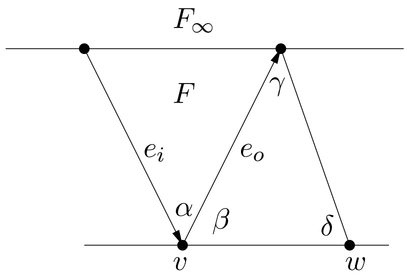

Figure 4.13: A vertex must be either a sink or a source.

Proof. Suppose that there is a triangle F and a vertex v such that F has two chordal edges e_i and e_o meeting at v, such that e_i is incoming and e_o is outgoing at v. See Figure 4.13. The colors of the angles α, β and γ must be pairwise different by Lemma 4.10. Hence, α and δ must have the same color (Lemma 4.9). But α has the same color as e_i. From Lemma 4.12 it follows that the alternating cycle (e_i, F_∞),(w, F) is monochromatic; contradiction. □

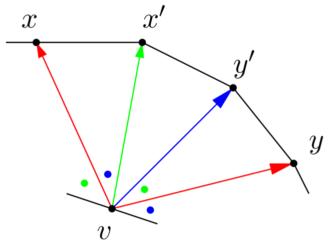

Figure 4.14: No two outgoing edges can have the same color.

Proposition 4.14. *No two outgoing edges from a vertex have the same color.*

Proof. Suppose that v has two outgoing edges of the same color and that all triangles are clockwise. From the coloring of angles it follows that edges sharing an angle have different colors. Even more, the colors of the outgoing edges at v in clockwise order cycle through 1,2,3. Hence, we find a sequence x, x', y', y of vertices, such that vx and vy have the same color, see Figure 4.14. Now, v is above x in blue and green, so any face incomparable to x which contains v has to be below x in red. The same is true for y. In particular $x > \{v, y', y\}$ and $y > \{v, x, x'\}$ in red. This is a monochromatic red alternating cycle; contradiction. □

Corollary 4.15. *No vertex belongs to four or more outgoing chordal edges.*

We say that the colors of the chordal edges bounding a face are *the colors of the face*. A face with two colors is called *bicolored*.

Proposition 4.16. *No two bicolored faces have the same colors.*

Proof. Suppose F and F' are two such faces. Suppose the two colors are red and green. Then F_∞ is below F and F' in red and green (Lemma 4.12). Therefore, F and F' can be below any vertex only in blue. Let x be vertex in $F \setminus F'$ and y be a vertex in $F' \setminus F$. Then (x, F'), (y, F) is a monochromatic blue alternating cycle; contradiction. □

We say that an oriented coloring satisfying Lemma 4.11, Proposition 4.14 and Proposition 4.16 is *permissible*. The map of Figure 4.15 is shown with a permissible coloring. We call it the *canonical map*. The vertices (edges) in the top of the figure are called q-vertices (edges) and the ones in the bottom of the figure are called p-vertices.

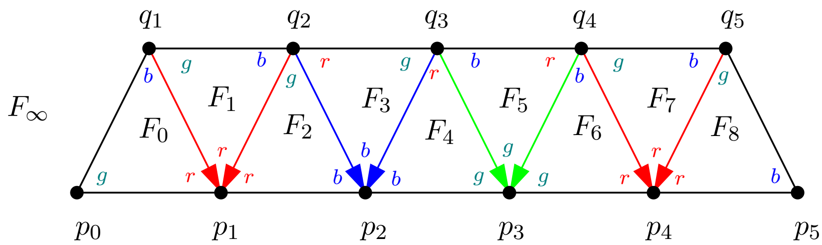

Figure 4.15: The canonical map.

Lemma 4.17. *Any maximal path-like map with a permissible coloring of the chordal edges can be constructed from the canonical map by a sequence of the following three operations:*

(i) Contracting a q-edge.

(ii) Subdividing a q-edge. Chordal edges between the new vertices and the p-vertex in the triangle are inserted with the same color and orientation as the old edges.

(iii) Deleting all the vertices, edges and faces on one side of a chordal edge.

Proof. Let M be a maximal path-like map with a permissible coloring of the chordal edges. If M has n sinks, then there are $n-1$ bicolored faces in M. From Proposition 4.16 and Lemma 4.11 it follows that the canonical map has the maximum possible number of sinks.

Again from Proposition 4.16 it follows that M has at most one vertex with outdegree 3. We can contract $\{q_2, q_3\}$ or $\{q_3, q_4\}$ in the canonical map to get such a vertex. Furthermore, any sinks or sources in the canonical map that are not in M can easily be removed using operation (iii). Hence, what remains is to possibly add some sources of degree one to the map M' we have constructed. But this is easy, since the new inserted chordal edges must have the same colors as the faces they split, i.e., we can just subdivide the q-edges as in (ii). □

Lemma 4.18. *Let M be the canonical map. Then* $\dim(\mathbf{P}_M) = 3$.

Proof. Consider the vertical symmetry through q_3 and the edge $\{p_2, p_3\}$. This partitions the vertex set into a left part $V_l = \{p_0, p_1, p_2, q_1, q_2\}$ a right part $V_r = \{\{p_5, p_4, p_3, q_5, q_4\}$ and $\{q_3\}$.

We construct three partial orders, one for each color. We start with the order of the vertices. In the green order we want $V_r > q_3 > V_l$. The ordering of the vertices in V_l and V_r is such that it allows to reverse all critical pairs corresponding to green angles, i.e., on $V_l : p_0 > q_1 > q_2 > \{p_2, p_1\}$ and on $V_r : p_3 > q_4 > q_5 > \{p_4, p_5\}$. We add the relations $p_2 > p_1$ and $p_5 > p_4$ so that the order on V_l and V_r conforms with the clockwise ordering around the outer face with start in p_0 and p_3, respectively.

The blue partial order is created symmetrically. I.e., it is obtained from the green order with the mappings $p_i \to p_{5-i}$ and $q_j \to q_{6-j}$. Note that the order on V_l and V_r conforms with the counterclockwise ordering around the outer face with start in p_2 and p_5, respectively.

For the red partial order, we construct two linear orders on $V_l \cup \{q_3\}$ and $V_r \cup \{q_3\}$. These linear orders gives us a partial order on $V_l \cup V_r \cup \{q_3\}$. In the linear order on $V_l \cup \{q_3\}$, the vertices come in the clockwise ordering around the outer face boundary with p_1 as maximal element, i.e, $p_1 > p_0 > q_1 > q_2 > q_3 > p_2$. The right part is done symmetrically, $p_4 > p_5 > q_5 > q_4 > q_3 > p_3$.

We now have three partial orders on the vertices. We extend these to partial orders on the vertices, edges and faces in three steps. First, we insert the Hamilton cycle edges and the outer face as low as possible in each of the three orders. Then the chordal edges are put above the outer face in their color, and as low as possible in the other two colors. Finally, the interior faces are inserted as low as possible.

Claim R. Every critical pair is reversed in one of the partial orders.

The lemma clearly follows from this claim; any three linear extensions of the partial orders constructed will then form a realizer.

There are three types of critical pairs: edge-face pairs, vertex-edge pairs and vertex-face pairs. All edge-face pairs are of the form (chordal edge, outer face), so they are reversed in the color of the edge.

Consider a vertex-edge critical pair (v, e). For the pair to be critical v and e must belong to a triangle and e has to be an edge of the Hamilton cycle. Such a critical pair corresponds to a colored angle at v. Since the order of each color reverses all critical pairs corresponding to this color each critical pair of this class is reversed.

It remains to prove that all vertex-face pairs (v, F) are reversed. Note that F is an interior face. If $v \in V_l$ and $F \subset V_r \cup \{q_3\}$, then (v, F) is reversed in blue. Similarly, if $v \in V_r$ and $F \subset V_l \cup \{q_3\}$, then (v, F) is reversed in green. All the vertices that are incomparable to F_4 are above F_4 in red, and q_3 is above all incomparable faces in green or blue. Hence, we only have to show that (v, F) is reversed when F and v are either both left or both right.

Suppose $v \in V_l$. The critical pairs (v, F_0) and $v = p_2, q_2$ are reversed in blue. The pair (p_0, F_1) is reversed in green and (p_2, F_1) in blue. The two pairs involving F_2 are reversed in green and all three pairs with F_3 in red. The cases where $v \in V_r$ are symmetric. □

Theorem 4.19. *Let M be a maximal path-like map. Then* $\dim(\mathbf{P}_M) = 3$ *if and only if there is a permissible coloring of the chordal edges.*

Proof. We have to prove that none of the operations of Lemma 4.17 increases the dimension.

(i): Only the q-edges $\{q_2, q_3\}$ and $\{q_3, q_4\}$ can be contracted (contracting one of the other q-edge is equivalent to the deletion of a part of the map). These two cases are symmetric, so we only consider the contraction of $\{q_2, q_3\}$. The new merged vertex $q_{2,3}$ takes the place of q_2 in green and blue, and the place of q_3 in red.

The vertex-edge pairs involving $q_{2,3}$ are reversed in green and red. Now, $q_{2,3}$ is only below the old position of q_2 in red. But the only vertex-face critical pair with q_2 that was reversed in red is (q_2, F_4), and $q_{2,3} \in F_4$, so all critical pairs involving $q_{2,3}$ are reversed.

Now, the position of a face in a partial order can only change if it contains q_2 or q_3 as its highest vertex. The new vertex $q_{2,3}$ is as high as q_2 in green and blue, and as high as q_3 in red, so the only affected face is F_5 in blue. In the blue partial order, the critical pairs (v, F_5), $v \in \{p_0, p_1, q_1\}$, are not reversed anymore. This is taken care of by moving q_1 (and hence p_1 and p_0) above q_4 in red. Since none of p_0, p_1 and q_1 where comparable to q_4 in red before, all previously reversed critical pairs are still reversed.

Suppose that both $\{q_2, q_3\}$ and $\{q_3, q_4\}$ are contracted. In this case it is enough to place the new vertex $q_{2,3,4}$ at the position of q_3 in all three vertex orders. The edges and face then are inserted by the above rules. This yields a realizer.

(ii): The partial orders are constructed as before (with possible changes resulting from a q-edge contraction). By an argument similar to the proof of Lemma 4.18, $\dim(\mathbf{P}_M) = 3$.

(iii): The only incidence that changes among the remaining elements of $\mathbf{P}_M$ is that one chordal edge e now is on the outer face. The edge e is moved below F_∞ in its color. The only new critical pair is (v, e), where v is the vertex in the same interior face as e that is not in e. But previously, there was a critical pair (v, F), $e \in F$, which was reversed, so (v, e) must be reversed. □

4.3.1 Algorithmic aspects

Theorem 4.19 can easily be turned into an algorithm for testing if $\dim(\mathbf{P}_M) \leq 3$ for a maximal path-like map M. Start by fixing the orientation and the color of one chordal edge. This induces an angle coloring in the adjacent triangles (Lemma 4.10, Lemma 4.9). Lemma 4.11 now gives us the colors of the angles in all the interior triangles in M, which in turns induces an oriented coloring of the chordal edges. Hence, given a fixed orientation of one chordal edge, any permissible coloring is unique up to permutations of the colors. To test for $\dim(\mathbf{P}_M) \leq 3$, we check if any vertex has four outgoing edges or if any two bicolored faces have the same colors. This can be done in linear time.

Once we have a permissible coloring of the chordal edges of M a 3-realizer can be generated: Since we now know which vertices are sinks, we know the

p-edges and q-edges of the colored map M and identify the operations of Lemma 4.17 (contracting a q-edge, subdividing a q-edge and removing a part of the map) that have to be applied to get M from the canonical map. The proof of Lemma 4.18 gives us a 3-realizer of the canonical map, and the proof of Theorem 4.19 demonstrates how to modify the 3-realizer for each of the operations. This yields an algorithm to produce a 3-realizer of $\mathbf{P}_M$ from a permissible coloring of M. It is clear from the proofs Lemma 4.18 and Theorem 4.19 that the running time of this algorithm can be bounded by some constant times the number of elements in $\mathbf{P}_M$. Since M is a planar map, we have the following theorem.

Theorem 4.20. *There is an algorithm running in time $O(n)$, which takes as input a maximal path-like map M with n vertices and either returns a 3-realizer of $\mathbf{P}_M$, or asserts that* $\dim(\mathbf{P}_M) = 4$.

4.4 Vertex-face posets of dimension at most 3

From Theorem 4.3 we know that if $\dim(\mathbf{Q}_M) \leq 3$, then M does not contain a subdivision of K_4. Figure 4.16 shows an example of a planar map which contains no K_4-subdivision but still $\dim(\mathbf{Q}_M) = 4$. This example from [15], has a dual map which is a $K_{2,3}$, with each edge replaced by a 2-face. More generally every map M where we can find fences of vertices and faces like in the proof of Theorem 4.5 must have $\dim(\mathbf{Q}_M) = 4$.

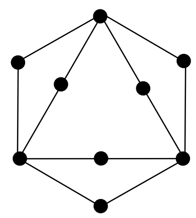

Figure 4.16: A planar map with vertex-face poset dimension 4.

However, unlike in the vertex-edge-face case, there are no 2-connected maps M of dimension 4 such that both M and M^* are weakly outerplanar.

Theorem 4.21. *Let M be a simple 2-connected planar map such both M and its dual M^* are weakly outerplanar. Then* $\dim(\mathbf{Q}_M) \leq 3$.

Proof. We may assume that no two vertices v, w of degree 2 are neighbors in M: Two such vertices are twin elements in $\mathbf{Q}_M$, hence contracting the edge v, w in the graph does not affect $\dim(\mathbf{Q}_M)$.

From Corollary 4.8, we know that if all chordal edges of M are moved inside the Hamilton cycle, the dual of the resulting map M' is $K_{2,3}$-subdivision free. Hence, M' is a path-like map. We will inductively construct two linear extensions, L_1 and L_2, of $\mathbf{Q}_M$, in which all vertex-face pairs are reversed. Similar to the proof of Proposition 4.7, we start with the path-like map $M_0 = M'$ and then move the required chordal edges outside the Hamilton cycle one by one, creating a series of maps $M_0, M_1, \ldots, M_k = M$.

Since M_0 is path-like, there must be exactly two faces that contains only one chordal edge. Each of these two faces contains a vertex of degree 2. Let these two vertices be ℓ and r.

Given a face F and a vertex x we say that *x is left of F* iff there is a x-ℓ path avoiding F. Symmetrically, *x is right of F* if there is a x-r path avoiding F. This definition of left of and right of coincides with the intuition of left and right based on a drawing where the Hamilton cycle is a circle, and ℓ and r are its left and right extreme points. Note that x is neither right, nor left of F if and only if $x \in F$. We define a vertex to be to the left (right) of a chordal edge in the same way.

Next, we inductively construct two linear extensions L_1^i and L_2^i of the vertex-face poset of M_i, for $i = 0, 1, \dots, k$, such that $L_1^k = L_1$ and $L_2^k = L_2$. Since M_0 is path-like, the interior dual of M_0 is a path. In the linear extensions L_1^0 and L_2^0 we order the interior faces by their position in this path, with the face containing ℓ highest in L_1^0 and the face containing r highest in L_2^0. The vertices and the outer face are inserted as high as possible.

Let e be the chordal edge that is moved outside the Hamilton cycle when M_i is changed to M_{i+1}. Before it is moved, it is contained in the two faces F' and F'' that are inside the Hamilton cycle. Let F' be the left one, i.e., let F' contain a vertex u that is left of e. When e is moved outside, some face G outside the Hamilton cycle is split into two faces G' and G'', let G' contain u. The faces F' and F'' are merged into a new face F^+ (see Figure 4.17).

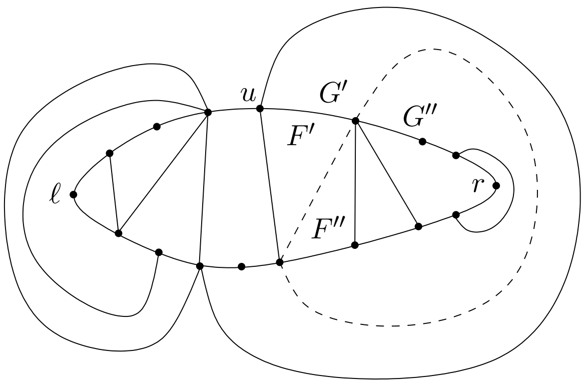

Figure 4.17: Moving the dashed edge from the inside to the outside.

We can now create L_1^{j+1} and L_2^{j+1} from L_1^j and L_2^j. In L_1^{j+1}, F^+ is inserted in the position of F' in L_1^j, G' is inserted in the position of G and G'' is inserted in the position of F''. In L_2^{j+1}, F^+ is inserted in the position of F'' in L_2^j, and G'' and G' are inserted in the positions of G and F', respectively.

Claim A. For $i = 0, 1, \dots, k$, L_1^i and L_2^i are linear extensions of $\mathbf{Q}_{M_i}$. If a vertex v is to the left of a face F in the map M_i, then $v > F$ in L_1^i, and if v is to the right of F, $v > F$ in L_2^i.

The claim can be verified by induction on i. From the construction of L_1^0 and L_2^0, it is clear that the claim is true for $i = 0$.

Suppose the claim is true for i. The map M^{i+1} is constructed by moving the edge e in M_i outside the Hamilton cycle. To verify that L_1^{i+1} and L_2^{i+1} are linear extensions of $\mathbf{Q}_M$ it is enough to check that F, G' and G'' are above all the vertices contained in them. This is immediate from the construction.

It remains to prove the second part of the claim. By induction and symmetry it is enough to consider the case where v is to the left of e and F is one of F^+, G' and G''. If v is to the left of e, either $v \in F'$ or v is to left of F'. In the latter case v is above F' in L_1^i, so $v > F^+$ in L_1^{i+1}. If v is to the left of G, then v is also to the left of G' and G''. Since $v > G$ in $L_1 i$, $v > G' > G''$ in L_1^{i+1}. On the other hand, if $v \in G$, v must also be in G' and to the left of G''. But v is to the left of F'', so $v > F''$ in L_1^i by construction, and hence $v > G''$ in L_1^{i+1}. Hence, the claim is true for $i + 1$. △

Claim A implies that all vertex-face critical pairs of $\mathbf{Q}_M$ are reversed in L_1 and L_2. It remains is to find a linear extension L_3 of $\mathbf{Q}_M$ which reverses all vertex-vertex and face-face critical pairs. A vertex-vertex pair (v, w) is critical when $w \in F$ only if $v \in F$ for all faces F, and a face-face pair (F, G) is critical when $v \in F$ only if $v \in G$ for all vertices v. Hence there is no alternating cycle containing only vertex-vertex pairs and face-face pairs. This implies that there is a linear extension L_3 reversing all these pairs. Together L_1, L_2, L_3 reverse all critical pairs. Hence, they form a realizer and $\dim(\mathbf{Q}_M) \leq 3$. □

In a strongly outerplanar map M it is never the case that we can find fences like in the proof of Theorem 4.5. The interior dual is a tree, and F_∞ contains all the vertices. Hence, F_∞^* has to be one of the degree 3-vertices of any $K_{2,3}$-subdivision in M^* (M contains no $K_{2,3}$-subdivision since it is outerplanar). Therefore, the existence of a strongly outerplanar map of vertex-face dimension 4 is not obvious.

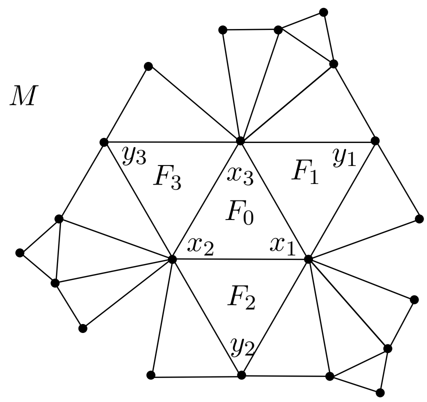

Figure 4.18: An outerplanar map with $\dim(\mathbf{Q}_M) = 4$.

Theorem 4.22. *The strongly outerplanar map M shown in Figure 4.18 has a vertex-face poset of dimension four.*

Proof. Suppose $\dim(\mathbf{Q}_M) \leq 3$. Again we identify the three linear extensions of a realizer with three colors and use these to color critical pairs. Our main focus will be on the coloring of critical pairs involving a vertex and an interior face.

Fact A. For an interior face F only two colors appear at critical pairs (v, F).

Suppose that the critical pair (F, F_∞) is reversed in color i. This forces all vertices below F in color i, and hence all critical pairs (v, F) are reversed in the other two colors. △

Fact B. If a triangular face $\Delta = \{v_1, v_2, v_3\}$ is surrounded by interior faces F_1, F_2, F_3 such that $v_i \notin F_i$, then the three critical pairs (v_i, F_i) use all three colors.

Any two of the three critical pairs form an alternating cycle and, hence, require different colors. Equivalently, the order induced by $v_1, v_2, v_3, F_1, F_2, F_3$ is a 3-crown. △

These two facts are applied to the central face F_0 of the map M: Fact B implies that the three critical pairs (x_i, F_{i-1}) use all three colors. Symmetry among the colors allows us to assume that (x_2, F_1) is red, (x_3, F_2) green and (x_1, F_3) blue. Fact A implies that two of the three critical pairs (y_i, F_0) have the same color. The symmetry of the graph allows to assume that this duplicated color is blue. It is infeasible to have (y_3, F_0) in blue, because it forms an alternating cycle with the blue pair (x_1, F_3). Hence, (y_1, F_0) and (y_2, F_0) are both blue.

To reach a contradiction we can from now on concentrate on the submap of M shown in Figure 4.19. The colors of critical pairs which have already been fixed are indicated by the colored arrows in the figure.

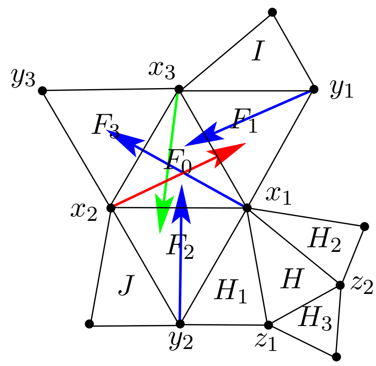

Figure 4.19: Having fixed the colors of some critical pairs, we concentrate on a submap M' of M.

To avoid a monochromatic alternating cycle with the blue pair (y_1, F_0) and with the green pair (x_3, F_2) the color of (x_1, I) has to be red. Similarly, the colors of (y_2, F_0) and (x_2, F_1) imply that (x_1, J) is green.

From the critical pairs (x_1, J) and (x_1, F_3) we know that $x_1 > x_2$ in green and blue, so all critical pairs (x_2, F), where $x_1 \in F$ must be red. Similarly, all critical pairs (x_3, F), where $x_1 \in F$ must be green. In particular we have (x_2, H_i) red and (x_3, H_i) green for $i = 1, 2$.

From Fact A applied to H_1 and H_2 we can conclude that neither (z_1, H_2) not (z_2, H_1) can be blue. Applying Fact B to face H we can conclude that (x_1, H_3) is blue. See Figure 4.20.

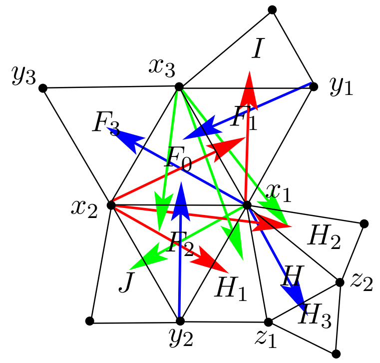

Figure 4.20: Colored critical pairs on M'.

Consider the critical pair (z_1, F_0). It forms an alternating cycle with (x_1, H_3), hence it can't be blue. It forms an alternating cycle with (x_2, H_1), hence it can't be red. It forms an alternating cycle with (x_3, H_1), hence it can't be green. Consequently there is no legal 3-coloring of the hypergraph of critical pairs of $\mathbf{Q}_M$.

□

The maximal outerplanar map T_4 shown in Figure 4.21 has a vertex-face poset of dimension 3 (a 3-realizer is listed in Table 4.1). Therefore, the example of a strongly outerplanar map M with $\dim(\mathbf{Q}_M) = 4$ given in Theorem 4.22 is close to a minimal example. Figure 4.22 shows a map where all 2-connected components are submaps of T_4 and hence have vertex-face poset dimension 3. Still an argument as in Theorem 4.22 shows that the map in Figure 4.22 has a 4-dimensional vertex-face poset.

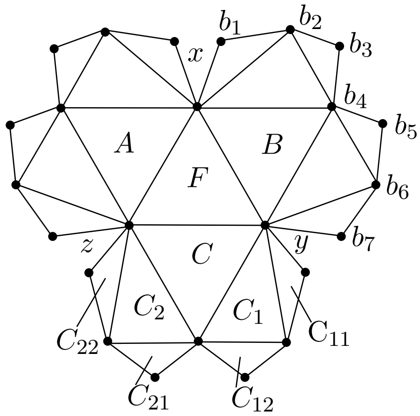

Figure 4.21: A 2-connected outerplanar map with vertex-face dimension 3. The naming scheme of the faces and vertices of the map is indicated in the figure.

4.5 Concluding remarks

When we started our investigations we set out to characterize the planar maps with vertex-edge-face posets of dimension at most 3. We proved that for all

L_1^1	L_2^1	L_3^1	L_1^2	L_2^2	L_3^2	L_1^3	L_2^3	L_3^3
A_{22}	C	C_{11}	B_2	F	c_7	c_7	a_1	b_7
A_2	B_{12}	C_1	b_4	C_{11}	a_2	C_2	A_1	B_2
A	B_{21}	C_{12}	B_{22}	c_1	a_1	C	A	B
F	B_2	C_{21}	b_6	C_1	a_3	z	z	y
B	B_{22}	C_2	b_7	y	a_5	C_{21}	A_{12}	B_{21}
B_1	F_∞	C_{22}	A_{21}	C_{12}	C	c_6	a_2	b_6
B_{11}	b_6	A_{11}	a_6	c_2	c_4	c_5	a_3	b_5
F_∞	b_3	A_1	a_5	c_3	A_{22}	C_{12}	A_{21}	B_{12}
x	b_5	A_{12}	A_{12}	C_{21}	a_7	c_3	a_5	b_3
a_7	b_7	A_{21}	a_3	c_5	A_2	C_1	A_2	B_1
b_1	B_{11}	F_∞	A_1	C_2	a_6	c_4	a_4	b_4
B_{12}	b_1	c_2	a_4	c_4	A	C_{11}	A_{22}	B_{11}
b_2	B_1	c_6	A_{11}	C_{22}	a_4	y	x	x
b_3	b_2	c_1	a_2	c_6	F	c_2	a_6	b_2
B_{21}	B	c_3	a_1	c_7	z	c_1	a_7	b_1
b_5	b_4	c_5	C_{22}	A_{11}	B_{22}			

Table 4.1: A 3-realizer of the map in Figure 4.21. The order L_i is obtained from the concatenation $L_i = L_i^1 \oplus L_i^2 \oplus L_i^3$.

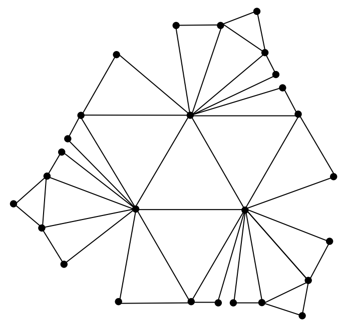

Figure 4.22: A map M with $\dim(\mathbf{Q}_M) = 4$, where each 2-connected component C has $\dim(\mathbf{Q}_C) = 3$

such maps M, both M and its dual M^* must be weakly outerplanar. In the case of maximal path-like maps, we found necessary and sufficient conditions for dimension at most 3 using an oriented coloring of the chordal edges. What remains open are four cases:

- M is path-like but not maximal, i.e., has non-triangular faces.

If M can be obtained from a maximal path-like map M_0 by subdividing some edges of the Hamilton cycle, then $\dim(\mathbf{P}_M) = \dim(\mathbf{P}_{M_0})$. Now, let M be a 2-connected map with $\dim(\mathbf{P}_M) \leq 3$. If a cycle edge $e = \{u, v\}$ is subdivided by a vertex w incident on the new cycle edges $e_1 = \{u, w\}$ and $e_2 = \{w, v\}$ to create a new map M', we can change a 3-realizer of $\mathbf{P}_M$ into a realizer of $\mathbf{P}_{M'}$ in the following way. In each linear extension, insert e_1 in the old position of e if $v < u$, and right below v if $v > u$. Symmetrically, e_2 is inserted into the position of e if $u < v$, and right below u if $u > v$. The vertex w is inserted right below $\min\{e_1, e_2\}$. This produces a 3-realizer of $\mathbf{P}_{M'}$. The general case remains a challenge.

- M is simple, and both M and M^* are weakly outerplanar, but neither of them is strongly outerplanar.

We can extend the coloring approach from Section 4.3 to get a set of necessary conditions for $\dim(\mathbf{P}_M) \leq 3$ in this case. There is a natural way to get an oriented coloring as in the case of path-like maps, if M is a different drawing of the graph of some maximal path-like map. Instead of coloring the angles of triangular faces, we color the angles of triangles in the strongly outerplanar drawing of the graph of M. Again, this angle coloring can be encoded as an oriented coloring of the chordal edges. Instead of each vertex being a sink or a source, each vertex will now be a sink on one side of the Hamilton cycle and a source on the other side. The proof of this is similar to the proof of Lemma 4.13.

If F is a face in the 2-connected map M with $\dim(\mathbf{P}_M) \leq 3$, the submap M_F induced by the vertices in F must be path-like by Theorem 4.5. In the same way, the submap $M^*_{v^*}$ of the dual map M^* induced by the dual vertices in the dual face v^* is also path-like. Since (x, y) is a critical pair in $\mathbf{P}_M$ if and only if (y^*, x^*) is a critical pair in $\mathbf{P}_{M^*}$, the primal oriented coloring induces an oriented coloring in the dual map. Hence, the oriented coloring of a map M with $\dim(\mathbf{P}_M) \leq 3$ must be permissible "locally" around each vertex and face. The question remains whether the existence of such a locally permissible coloring is also a sufficient condition for $\dim(\mathbf{P}_M) \leq 3$, or if there are some non-local effects that force $\dim(\mathbf{P}_M) = 4$.

- M is not simple.
- M is not 2-connected.

Suppose M is not 2-connected. From $\dim(\mathbf{P}_C) \leq 3$ for each 2-connected component C it can not be concluded that $\dim(\mathbf{P}_M) \leq 3$. The conclusion is not even possible if all components are maximal path-like maps and have a common outer face. Consider the map M constructed by taking two maximal path-like maps C_1 and C_2 and identifying two vertices $v_1 \in C_1$ and $v_2 \in C_2$ and the outer faces of each map. We choose C_1 and C_2 such that $\dim(\mathbf{P}_{C_1}) = \dim(\mathbf{P}_{C_2}) = 3$ and that in any permissible coloring of the chordal edges in each map C_i there will be two outgoing edges from v_i. Such maps clearly exist. A straightforward modification of Proposition 4.14 will now show that $\dim(\mathbf{P}_M) = 4$.

Vertex-face posets and posets of height 2

For vertex-face posets, we saw that it seems hard to characterize even the strongly outerplanar 2-connected maps with dimension at most 3. This relates to the long-standing open question if it is NP-hard to determine if the dimension of a height 2 poset is at most 3. Yannakakis [60] proved in 1982 that it is NP-hard to determine if $\dim(\mathbf{P}) \leq 3$ for posets $\mathbf{P}$ of height at least 3. Brightwell and Trotter [15] refined the question and asked if it is NP-hard to recognize planar maps with $\dim(\mathbf{Q}_M) \leq 3$. Given our results, it makes sense to ask this question even for 2-connected strongly outerplanar maps.

Chapter 5

Approximation algorithms for graphs with large treewidth

5.1 Introduction

One of the most successful parameterizations of graphs is that of *treewidth*. While the formal definition is deferred to the next section, graphs of treewidth k, also known as *partial k-trees*, are graphs that admit a tree-like structure, known as their *tree-decomposition* of *width* k.

A wide variety of NP-hard graph problems have been shown to be solvable in polynomial time, or even linear time, when constrained to partial k-trees [6, 7, 56]. For some of these problems polynomial time solutions are possible for graphs of treewidth $O(\log n)$ or $O(\log n/\log\log n)$ [7, 56].

A standard example of a problem solvable in graphs of treewidth $O(\log n)$ is the maximum independent set (MIS) problem [7], which is that of finding a maximum collection of pairwise non-adjacent vertices. In the weighted version of the problem, vertices are given with weights and we seek an independent set of maximum total weight. For general graphs, the best polynomial-time approximation ratios known for MIS is $n\,(\log\log n)^2/\log^3 n$ [29]. On the other hand, it is known that unless NP $\subseteq$ ZPTIME$(2^{(\log n)^{O(1)}})$, no polynomial-time algorithm can achieve an approximation guarantee of $n^{1-O(1/(\log n)^{\gamma})}$ for some constant γ [39].

In this chapter, we investigate the approximability status of some of the aforementioned NP-hard problems, where our main interest is in graphs of treewidth $k = \omega(\log n)$. We focus our study on MIS, deriving further applications of our method by extensions of that given for MIS.

Better approximation bounds for MIS are achievable for special classes of graphs. For the purposes of this chapter, a class that properly contains partial k-trees is that of *k-inductive graphs*. A graph is said to be k-inductive if there is an ordering of its vertices so that each vertex has at most k higher-numbered neighbors. If such an ordering exists, it can be found by iteratively choosing and removing a vertex of minimum degree in the remaining graph. From this definition, it is clear that k-trees, and thus also partial k-trees, are k-inductive. A k-inductive graph is easily $k+1$-colored by processing the vertices in their reverse inductive order, assigning each vertex one of the colors not used by its at most k previously colored neighbors. This implies that the largest weight color class approximates the weighted MIS within a factor of $k+1$. The best approximation known for MIS (and weighted MIS) in k-inductive graphs is $O(k \log\log k/\log k)$ [35].

5.1.1 New contribution

We present a novel generic scheme for approximation algorithms for maximum independent set and other NP-hard graph optimization problems constrained to graphs of treewidth $k = \omega(\log n)$. Our scheme leads to deterministic polynomial-time algorithms that achieve an approximation ratio of $\ell/\log n$ when a tree-decomposition of width $\ell = \Omega(\log n)$ is given.

Our scheme can be applied to any problem of finding a maximum induced subgraph with hereditary property Π and any problem of finding a minimum partition into induced subgraphs with hereditary property Π provided that for graphs with given tree-decomposition of logarithmic or near logarithmic width can be solved exactly in polynomial time. All these approximation factors achievable in polynomial time are the best known for the aforementioned problems for graphs of superlogarithmic treewidth (see Table 5.1 for some examples).

Problem	Old	New
Max independent set	$O(k \log\log k/\log k)$ [35]	$O(k/\log n)$
Max clique	$O(n(\log\log n)^2/\log^3 n)$ [29]	$O(\min(k/\log k,$ $k(\log\log k)^2/\log^3 k))$
Min vertex coloring	$O(n^{1-3(k+1)} \log n)$ [38]	$O(k \log\log k/\log k)$

Table 5.1: Examples of old and new approximation ratios for optimization problems in graphs with treewidth k (assuming a tree-decomposition of width $O(k)$ is given).

In case a tree-decomposition of width $\ell = k$ is not given, the approximation achieved by our method increases by a factor of $O(\sqrt{\log k})$.

5.2 Preliminaries

The notion of *treewidth* of a graph was originally introduced by Robertson and Seymour [50] in their seminal graph minors project. It has turned out to be equivalent to several other interesting graph theoretic notions, e.g., that of partial k-trees.

Definition 5.1. A *tree-decomposition* of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$, where $\{X_i \mid i \in I\}$ is a collection of subsets of V, and $T = (I, F)$ is a tree, such that the following three conditions are fulfilled:

1. $\bigcup_{i \in I} X_i = V$,

2. for each edge $(v, w) \in E$, there exists a node $i \in I$, with $v, w \in X_i$, and

3. for each vertex $v \in V$, the subgraph of T induced by the nodes $\{i \in I \mid v \in X_i\}$ is connected.

The *size of* T is the number of nodes in T, that is, $|I|$. Each set X_i, $i \in I$, is called the *bag* associated with the ith node of T. The width of a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$. The *treewidth* of a graph is the minimum width of its tree-decomposition taken over all possible tree-decompositions of the graph.

It is well known that a graph G is a partial k-tree iff the treewidth of G is at most k [6]. For a graph with n vertices and treewidth k, a tree decomposition of width k can be found in time $O(n\, 2^{O(k^3)})$ [11], whereas a tree decomposition of width $O(k\sqrt{\log k})$ and size $O(n)$ can be found in time polynomial in n [30].

For technical reasons, it will be more convenient to use a special form of tree-decomposition termed as *nice tree-decomposition*.

Definition 5.2. A tree-decomposition $T = (I, F)$ of a graph G is *nice* if

1. T is a binary rooted tree,

2. if a node $i \in I$ has two children j_1 and j_2, then $X_i = X_{j_1} = X_{j_2}$ (i is called a *join* node),

3. if a node $i \in I$ has one child j, then either $X_j \subset X_i$ and $|X_i - X_j| = 1$, or $X_i \subset X_j$ and $|X_j - X_i| = 1$ (i is called an *introduce* or a *forget* node, respectively).

Fact 5.3. [43] *A tree-decomposition $T = (I, F)$ of a graph G can be transformed into a nice tree-decomposition in time polynomial in $|I|$ and the size of T, without increasing its width. The size of the resulting nice decomposition is $O(\ell \cdot |I|)$, where ℓ is the width of T.*

5.3 Approximation of maximum independent set

In this section we present a deterministic approximation algorithm for finding maximum independent set in graphs with given tree-decomposition of width ℓ. We begin with the following general partition lemma of independent interest.

Lemma 5.4. *Let t be a positive integer and let G be a graph given with a tree decomposition T of width ℓ. Then, the vertex set of G can be partitioned in polynomial time into classes $V_1, \ldots, V_{\lceil (\ell+1)/t \rceil}$ so that each bag of T contains at most t vertices from each class.*

Proof. By Fact 5.3, we may assume w.l.o.g. that the given tree decomposition T is nice.

To obtain the intended partition, we proceed top-down, arbitrarily assigning the vertices in the root-bag of T into classes with at most t vertices each. Inductively, for a node v with a child u in T, the partition of the bag of u is consistent with that of the bag of v and the upper bound of t on the size of each class V_i within each bag. Namely, if v is a join node, the bag of u gets the same partition as that of v; if v is an introduce node, the partition is also the same (with one fewer vertices); if v is a forget node, then the additional vertex in u is placed into some class that has fewer than t vertices from the bag of u. □

Lemma 5.4 yields the aforementioned approximation algorithm for maximum independent set.

Theorem 5.5. *Let c be a positive constant. For a graph G on n vertices given with its tree-decomposition of width ℓ and of polynomial size, the maximum weighted independent set problem admits a $\lceil(\ell+1)/(c\log n)\rceil$-approximation algorithm.*

Proof. Apply Lemma 5.4 to T with $t = c\log n$ to obtain a vertex partition $V_1, \ldots, V_{\lceil(\ell+1)/(c\log n)\rceil}$.

For any i, $1 \le i \le \lceil(\ell+1)/(c\log n)\rceil$, let G_i be the subgraph of G induced by the vertex set V_i and let T_i be the tree-decomposition of G_i obtained by constraining the bags of T to the vertices in V_i. By the properties of the classes V_i, each T_i has width at most $c\log n$.

For each i, $1 \le i \le \lceil(\ell+1)/(c\log n)\rceil$, we can find a maximum independent set in G_i by using the standard dynamic programming method on T_i [7]. By the pigeon hole principle, at least one of these satisfies the claim of theorem. □

5.4 Extensions of the approximation method

We can generalize Theorem 5.5 to include the problem of *maximum weight induced subgraph with hereditary property* Π provided that the problem constrained to graphs of treewidth $O(\log n)$ can be solved exactly in polynomial time. For a graph with vertex weights, the problem of maximum weight induced subgraph with property Π is to find a maximum weight subset of vertices of the input graph which induces a subgraph having the property Π. If Π holds for arbitrarily large graphs, does not hold for all graphs, and is *hereditary* (holds for all induced subgraphs of a graph whenever it holds for the graph) then the problem of finding a maximum weight induced subgraph with the property Π is NP-hard (see GT21 in [33]). Examples of such properties Π are "being an independent set," "being m-colorable," and "being a planar graph."

Theorem 5.6. *For a graph G given with tree-decomposition of width $\ell \ge t > 0$ and of polynomial size, the problem of finding a maximum weight induced subgraph with hereditary property Π admits a $\lceil(\ell+1)/t\rceil$-approximation, provided that for graphs of treewidth t the problem can be solved exactly in polynomial time.*

Proof. Similar to Theorem 5.5. □

Since for every clique W in a graph G and every tree decomposition $(\{X_i | i \in I\},\ T)$ of G, there is an $i \in I$ with $W \subseteq X_i$, it suffices to check each subset in each bag of the given tree decomposition in order to find a maximum weight clique. Hence, we can approximate the clique problem by applying any clique-approximation algorithm on the graphs induced by each X_i. E.g., we can check each subset of logarithmic size or apply the algorithm from [29]. Combined with Theorem 5.6, we obtain the following corollary.

Corollary 5.7. *Let c be any positive constant. For a graph G on n vertices given with its tree-decomposition of width $\ell \geq c \log n$ and of polynomial size, the problem of maximum weighted clique admits a* $\min((\ell+1)/(c\log n), \ell(\log\log \ell)^2/\log^3 \ell)$*-approximation.*

The problem of *minimum partition into induced subgraphs with property* Π is to find a minimum cardinality partition of vertices of the input graph into subsets inducing subgraphs having the property Π. E.g., if Π is "being independent," we get the minimum coloring problem.

Theorem 5.8. *For a graph G on n vertices given with its tree-decomposition of width $\ell \geq t > 0$ and of polynomial size, the problem of minimum partition into induced subgraphs with hereditary property* Π *admits a* $\lceil(\ell+1)/t\rceil$*-approximation algorithm, provided that for $\ell = t-1$ the problem can be solved exactly in polynomial time.*

Proof. Produce the subgraphs G_i, $1 \leq i \leq \lceil(\ell+1)/t\rceil$, as in the proof of Theorem 5.5. For each G_i find a minimum number partition P_i into induced subgraphs with hereditary property Π and output the union of P_i as the approximate solution. □

Since the minimum vertex coloring problem can be solved exactly in polynomial time for graphs with given tree-decomposition of width $O(\log n/\log\log n)$ [56], we obtain the following.

Corollary 5.9. *Let c be any positive constant. For a graph G on n vertices given with its tree-decomposition of width ℓ and of polynomial size, the minimum vertex coloring problem admits* $\lceil(\ell+1)\log\log n/(c\log n)\rceil$*-approximation.*

Since a tree-decomposition of width $O(k\sqrt{\log k})$ and size $O(n)$ can be found in time polynomial in n [30], we obtain the following variants of Theorems 5.6 and 5.8 for graphs of treewidth k.

Theorem 5.10. *Let $k \geq t > 0$ and let G be a graph with treewidth k. The problems of maximum weight induced subgraph with hereditary property* Π *and the problems of minimum partition into induced subgraphs with hereditary property* Π *admit* $\sqrt{\log k}\lceil(k+1)/t\rceil$*-approximation algorithm provided that for a graph of treewidth $t-1$ they can be solved in polynomial time.*

In [56], classes of vertex partitioning problems that can be solved in polynomial time on graphs of near logarithmic treewidth are given. Thus, for problems in these classes, Theorems 5.8 and 5.10 can be used.

Notation

General

$\|A\|$	cardinality of the set A
log	binary logarithm

Graphs and maps

G	graph
$V(G)$	set of vertices in the graph G
$E(G)$	set of edges in the graph G
$G[U]$	graph induced by the vertex (edge) subset $U \subseteq V(G)$ ($U \subseteq E(G)$)
$G \setminus U$	subgraph of G induced by $V(G) \setminus U$
$H + R$	subgraph H and subgraph R together with H-R edges in a graph G
$\{u, v\}$	undirected edge
(u, v)	directed edge from u to v
M	planar map
M^*	the dual map of M

Posets

$\mathbf{P}$	poset
$\mathbf{P}_{V(G)}$	poset induced by $V(G)$, G acyclic digraph
$\mathbf{P}_M$	vertex-edge-face poset of the map M
$\mathbf{Q}_M$	vertex-face poset of M
(a, b)	critical pair
$\dim(\mathbf{P})$	dimension of the poset $\mathbf{P}$
$a\|\|b$	a and b are incomparable.

Bibliography

[1] A. V. Aho, M. R. Garey, and J. D. Ullman, *The transitive reduction of a directed graph*, SIAM J. Comput. **1** (1972), no. 2, 131–137.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data structures and algorithms*, Addison Wesley, 1983.

[3] S. Alstrup, C. Gavoille, H. Kaplan, and T. Rauhe, *Nearest common ancestors: A survey and a new algorithm for a distributed environment*, Theor. Comput. Syst. **37** (2004), 441–456.

[4] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares, *On sparse spanners of weighted graphs*, Discrete Comput. Geom. **9** (1993), 81–100.

[5] S. R. Arikati, D. Z. Chen, L. P. Chew, G. Das, M. H. M. Smid, and C. D. Zaroliagis, *Planar spanners and approximate shortest path queries among obstacles in the plane*, Proceedings of the 4th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci., vol. 1136, Springer-Verlag, 1996, pp. 514–528.

[6] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, J. Algorithms **12** (1991), no. 2, 308–340.

[7] S. Arnborg and A. Proskurowski, *Linear time algorithms for NP-hard problems restricted to partial k-trees*, Discrete Appl. Math. **23** (1989), no. 1, 11–24.

[8] L. Babai and D. Duffus, *Dimension and automorphism groups of lattices*, Algebra Universalis **12** (1981), no. 3, 279–289.

[9] S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie, *New constructions of (α, β)-spanners and purely additive spanners*, Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2005, pp. 672–681.

[10] S. Baswana and S. Sen, *A simple linear time algorithm for computing a $(2k-1)$-spanner of $O(n^{1+1/k})$ size in weighted graphs*, Proceedings of the 30th Annual International Colloquium on Automata, Languages and Programming (ICALP 2003), Lecture Notes in Comput. Sci., vol. 2719, Springer-Verlag, 2003, pp. 384–296.

[11] H. L. Bodlaender, *A linear time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput. **25** (1996), no. 6, 1305–1317.

[12] B. Bollobás, D. Coppersmith, and M. Elkin, *Sparse distance preservers and additive spanners*, SIAM J. Discrete Math. **19** (2005), no. 4, 1029–1055.

[13] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph classes: a survey*, SIAM Monogr. on Discrete Math. and Appl., vol. 3, Society for Industrial and Applied Mathematics.

[14] G. R. Brightwell and W. T. Trotter, *The order dimension of convex polytopes*, SIAM J. Discrete Math. **6** (1993), no. 2, 230–245.

[15] ______, *The order dimension of planar maps*, SIAM J. Discrete Math. **10** (1997), no. 4, 515–528.

[16] J. Chang, C. Ho, and M. Ko, *LexBFS-ordering in asteroidal triple-free graphs*, ISAAC '99: Proceedings of the 10th International Symposium on Algorithms and Computation, Lecture Notes in Comput. Sci., vol. 1741, Springer Verlag, 1999, pp. 163–172.

[17] D. Z. Chen and J. Xu, *Shortest path queries in planar graphs*, STOC '00: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, ACM Press, 2000, pp. 469–478.

[18] E. Cohen, *Fast algorithms for constructing t-spanners and paths with stretch t*, SIAM J. Comput. **28** (1998), 210–236.

[19] D. Coppersmith and M. Elkin, *Sparse source-wise and pair-wise distance preservers*, Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2005, pp. 660–669.

[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 2nd ed., MIT Press, 2001.

[21] R. Diestel, *Graph theory*, third ed., Springer Verlag, Heidelberg, 2005.

[22] H. Djidjev, *Efficient algorithms for shortest path queries in planar digraphs*, Proceedings of the 22nd International Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci., vol. 1197, Springer-Verlag, 1996.

[23] H. Djidjev, G. E. Pantziou, and C. D. Zaroliagis, *Fast algorithms for maintaining shortest paths in outerplanar and planar digraphs*, Proceedings of the 10th International Symposium on Fundamentals of Computation Theory, Lecture Notes in Comput. Sci., vol. 965, Springer-Verlag, 1995, pp. 191–200.

[24] B. Dushnik and E. Miller, *Partially ordered sets*, Amer. J. Math. **63** (1941), 600–610.

[25] M. Elkin and D. Peleg, *Strong inapproximability of the basic k-spanner problem*, Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP 2000), Lecture Notes in Comput. Sci., vol. 1853, Springer-Verlag, 2000, pp. 636–647.

[26] ———, *$(1+\epsilon,\beta)$-spanner constructions for general graphs*, SIAM J. Comput. **33** (2004), no. 3, 608–631.

[27] M. Elkin and J. Zhang, *Efficient algorithms for constructing $(1+\epsilon,\beta)$-spanners in the distributed and streaming models*, Proceedings of the 23rd Annual ACM Symposium on Principles of Distributed Computing, ACM Press, 2004, pp. 160–168.

[28] D. Eppstein, Z. Galil, G. F. Italiano, and T. H. Spencer, *Separator-based sparsification II: Edge and vertex connectivity*, SIAM J. Comput. **28** (1999), no. 1, 341–381.

[29] U. Feige, *Approximating maximum clique by removing subgraphs*, SIAM J. Discrete Math. **18** (2004), no. 2, 219–225.

[30] U. Feige, M. Hajiaghayi, and J. R. Lee, *Improved approximation algorithms for minimum-weight vertex separators*, STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, ACM Press, 2005, pp. 563–572.

[31] S. Felsner and W. T. Trotter, *Posets and planar graphs*, J. Graph Theory **49** (2005), no. 4, 273–284.

[32] S. Felsner and F. Zickfeld, *On the number of α-orientations*, Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci., Springer Verlag, 2007, to appear.

[33] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, New York, NY, 1979.

[34] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz, *Distance labeling in graphs*, J. Algorithms **53** (2004), no. 1, 85–112.

[35] M. M. Halldórsson, *Approximations of weighted independent set and hereditary subset problems*, J. Graph Algorithms Appl. **4** (2000), no. 1.

[36] H. T. Hsu, *An algorithm for finding a minimal equivalent graph of a digraph*, J. ACM **22** (1975), no. 1, 11–16.

[37] T. Kameda, *On the vector representation of the reachability in planar directed graphs*, Inform. Process. Lett. **3** (1974/75), 75–77.

[38] D. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring by semidefinite programming*, J. ACM **45** (1998), no. 2, 246–265.

[39] S. Khot, *Improved inapproximability results for MaxClique, chromatic number, and approximate graph coloring*, Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, 2001, pp. 600–609.

[40] S. Khuller, B. Raghavachari, and N. E. Young, *Approximating the minimum equivalent digraph*, Proceedings of the 5th annual ACM-SIAM Symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 1994, pp. 177–186.

[41] P. N. Klein, *A subset spanner for planar graphs, with application to subset TSP*, STOC '06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing, ACM Press, 2006, pp. 749–756.

[42] P. N. Klein and S. Subramanian, *A fully dynamic approximation scheme for shortest paths in planar graphs*, Algorithmica **22** (1998), 235–249.

[43] T. Kloks, *Treewidth: Computations and approximations*, Lecture Notes in Computer Science, vol. 842, Springer-Verlag, Heidelberg, 1994.

[44] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math. **36** (1979), no. 2, 177–189.

[45] S. L. Mitchell, *Linear algorithms to recognize outerplanar and maximal outerplanar graphs*, Inform. Process. Lett. **9** (1979), no. 5, 229–232.

[46] R.H. Möhring, *Computationally tractable classes of ordered sets*, Algorithms and Orders (I. Rival, ed.), Reidel Dordrecht, 1989, pp. 105–194.

[47] D. Peleg, *Proximity-preserving labeling schemes*, J. Graph Theory **33** (2000), no. 3, 167–176.

[48] D. Peleg and A. A. Schaffer, *Graph spanners*, J. Graph Theory **13** (1989), 99–116.

[49] O. Reingold, *Undirected ST-connectivity in log-space*, STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, ACM Press, 2005, pp. 376–385.

[50] N. Robertson and P. Seymour, *Graph minors. II. Algorithmic aspects of tree-width*, J. Algorithms **7** (1986), no. 3, 309–322.

[51] L. Roditty, M. Thorup, and U. Zwick, *Roundtrip spanners and roundtrip routing in directed graphs*, Proceedings of the 13th Annual ACM-SIAM Symposium On Discrete Algorithms, Society for Industrial and Applied Mathematics, 2002, pp. 844–851.

[52] W. J. Savitch, *Relationships between nondeterministic and deterministic tape complexities*, J. Comput. Syst. Sci. **4** (1970), no. 2, 177–192.

[53] W. Schnyder, *Planar graphs and poset dimension*, Order **5** (1989), 323–343.

[54] S. Subramanian, *A fully dynamic data structure for reachability in planar digraphs*, Proceedings of the 1st Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci., vol. 726, Springer-Verlag, 1993, pp. 372–383.

[55] R. Tamassia and I. G. Tollis, *Dynamic reachability in planar digraphs with one source and one sink*, Theor. Comput. Sci. **119** (1993), no. 2, 331–343.

[56] J. A. Telle and A. Proskurowski, *Algorithms for vertex partitioning problems on partial k-trees*, SIAM J. Discrete Math. **10** (1997), no. 4, 529–550.

[57] M. Thorup, *Compact oracles for reachability and approximate distances in planar digraphs*, J. ACM **51** (2004), no. 6, 993–1024.

[58] W. T. Trotter, *Combinatorics and partially ordered sets*, Johns Hopkins Series in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1992.

[59] ———, *Graphs and partially ordered sets: recent results and new directions*, Congr. Numer. **116** (1996), 253–278.

[60] M. Yannakakis, *The complexity of the partial order dimension problem*, SIAM J. Algebraic Discrete Methods **3** (1982), no. 3, 351–358.

Index