Decompositions and Algorithms for the Disjoint Paths Problem in Planar Graphs

Giannos Stamoulis
R.N.: A0026

Examination committee:
Michael C. Dracopoulos, Department of Mathematics, National and Kapodistrian University of Athens.
Archontia C. Giannopoulou, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.
Stavros G. Kolliopoulos, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.

Supervisor:
Stavros G. Kolliopoulos, Professor, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.

Μεταπτυχιακό Πρόγραμμα «Αλγόριθμοι, Λογική και Διακριτά Μαθηματικά» – 2016

ABSTRACT

In the Disjoint Paths Problem, given a graph G and a set of k pairs of terminals, we ask whether the pairs of terminals can be linked by pairwise disjoint paths. In the *Graph Minors series* of 23 papers between 1984 and 2011, Neil Robertson and Paul D. Seymour, among other great results that heavily influenced Graph Theory, provided an $f(k) \cdot n^3$ algorithm for the Disjoint Paths Problem. To achieve this, they introduced the *irrelevant vertex technique* according to which in every instance of treewidth greater than $g(k)$ there is an "irrelevant" vertex whose removal creates an equivalent instance of the problem.

We study the problem in the case of planar graphs and we prove that for every fixed k every instance of the Planar Disjoint Paths Problem can be transformed to an equivalent one that has bounded treewidth, by simultaneously discarding a set of vertices of the given planar graph. As a consequence the Planar Disjoint Paths Problem can be solved in linear time for every fixed number of terminals.

ΣΥΝΟΨΗ

Στο πρόβλημα των Διακεκριμενων Μονοπατιων μας ζητείται να εξετάσουμε, δεδομένου ενός γραφήματος G και ενος συνόλου k ζευγών τερματικών, αν τα ζεύγη των τερματικών μπορούν να συνδεθούν με διακεκριμένα μονοπάτια. Στα "Graph Minors", μια σειρά 23 εργασιών μεταξύ 1984 και 2011, οι Neil Robertson και Paul D. Seymour, ανάμεσα σε άλλα σπουδαία αποτελέσματα που επηρέασαν βαθιά την Θεωρία Γραφημάτων, παρουσίασαν έναν $f(k)\cdot n^3$ αλγόριθμο για το πρόβλημα των Διακεκριμενων Μονοπατιων. Για να το καταφέρουν αυτό, εισήγαγαν την *"τεχνκή της άσχετης κορυφής"* σύμφωνα με την οποία σε κάθε στιγμιότυπο δεντροπλάτους μεγαλύτερου του $g(k)$ υπάρχει μια "άσχετη" κορυφή της οποίας η αφαίρεση δημιουργεί ένα ισοδύναμο στιγμιότυπο του προβλήματος.

Εδώ μελετάμε το πρόβλημα σε επίπεδα γραφήματα και αποδεικνύουμε ότι για κάθε σταθερό k κάθε στιγμιότυπο του προβλήματος των Διακεκριμενων Μονοπατιων σε επιπεδα γραφηματα μπορεί να μετασχηματιστεί σε ένα ισοδύναμο που έχει φραγμένο δενδροπλάτος, αφαιρώντας ταυτόχρονα ένα σύνολο κορυφών από το δεδομένο επίπεδο γράφημα. Ως συνέπεια αυτού, το πρόβλημα των Διακεκριμενων Μονοπατιων σε επιπεδα γραφηματα μπορεί να λυθεί σε γραμμικό χρόνο για κάθε σταθερό πλήθος τερματικών.

ACKNOWLEDGEMENTS

Foremost, I would like to express my deep gratitude to my advisor, Professor Stavros G. Kolliopoulos, for his valuable guidance, encouragement and helpful critique.

I would also like to give my sincere thanks to the other members of the examination committee, Assistant Professor Archontia C. Giannopoulou and Assistant Professor Michael C. Dracopoulos, for generously providing their time and support.

Lastly, I would like to express my gratefulness to my friends and family for providing me scientific, artistic and emotional support not only while working on this thesis but throughout my entire presence in this postgraduate program.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 The Disjoint Paths Problem

One central question in Graph Theory, from the algorithmic point of view, is whether two vertices u, v of a given graph G are connected, i.e., if there exists a path of G with u, v as its endpoints. This problem is known as REACHABILITY and in its formal statement is the following:

> REACHABILITY
> *Input*: A graph G, and two vertices $u, v \in V(G)$.
> *Question*: Is there a path in G with endpoints u, v?

It is well-known that REACHABILITY admits polynomial-time algorithms such as breadth-first search and depth-first search. Issues arise when we consider multiple pairs of vertices of a given graph and ask whether there exist paths linking each pair in G. These vertices are often called *terminals*. If this question does not place any limitation on how these paths intersect then we can easily notice that we can use one of the aformentioned algorithms for REACHABLITY. But what happens if we demand our paths to be edge-disjoint or vertex-disjoint, i.e., two or more paths do not share an edge or a vertex, respectively? We focus here on the vertex-disjoint version on the problem:

> DISJOINT PATHS (DPP)
> *Input*: A graph G, and a set $\mathcal{T} = \{(s_i, t_i) \in V(G)^2, i \in \{1, \ldots, k\}\}$ of pairs of terminals of G.
> *Question*: Are there k pairwise vertex-disjoint paths $P_1, \ldots, P_k$ in G such that for $i \in \{1, \ldots, k\}$, P_i has endpoints s_i and t_i?

The DISJOINT PATHS problem (in short DPP), as well its directed and edge-disjoint variants, have attracted a lot of research. This is not only because of the numerous applications in network routing, in transportation, and in VLSI design but also because it inspired a lot of research in graph algorithms and combinatorial optimization (see [8–10, 13, 31]).

From the scope of computational complexity, Karp showed in [15] that DPP is NP-complete. Also, later it was proved that DPP remains NP-complete even when the input graph is restricted to be a planar graph [22] as well as in other variants of the problem (see [20, 24, 35]).

But what happens when we are given a graph G and we are asked whether there exist two vertex-disjoint paths connecting two given pairs of vertices? The answer is that there exist polynomial time algorithms that solve the problem as those presented independently in [32–34] in 1980.

An important breakthrough in the algorithmic study of DPP was achieved by Roberson and Seymour in [26]. Given that the number of pairs of terminals is a fixed number that is not part of the input but instead is given as a parameter, the algorithm in [26] solves the DPP in $O(n^3)$ steps. As an important ingredient of this algorithm Robertson and Seymour introduced in [26] the *irrelevant vertex technique*. This technique asks for structural characteristics of the input of a problem on graphs that may permit the detection, in polynomial time, of a non-terminal vertex v in G such that $(G, \mathcal{T})$ and $(G \setminus v, \mathcal{T})$ are equivalent instances, i.e., they are either both yes-instances or both no-instances of the problem.

The irrelevant vertex technique has nowadays evolved to a standard algorithmic paradigm for solving problems that are related to the identification of paths or collections of paths in graphs [1, 2, 7, 11, 12, 14, 17, 19, 23].

In the case of [26], the structural characteristic that permitted the application of the irrelevant vertex technique was the presence of a "big-enough" clique in G as a minor or, provided that such a clique does not exist, the presence of a "big-enough" grid as a minor (see [26–28] for the justification of these conditions). Given these two combinatorial facts, after successively removing irrelevant vertices, we end up with an equivalent DPP-instance whose graph G excludes a grid as a minor. This in turn implies that G has "small-enough" treewidth and thus the problem can be solved in linear time, using dynamic programming techniques. As the detection of an irrelevant vertex in [26] requires $O(n^2)$ steps and at most n irrelevant vertices can be discarded, the overall running time of the algorithm is $O(n^3)$. This running time was improved by Kawarabayashi, Kobayashi, and Reed in [16] who derived an $O(n^2)$ step algorithm by giving procedures, alternative to those of [26], that can detect irrelevant vertices in linear time.

An interesting question in all the aforementioned algorithms is the contribution of the parameter k in the "O"-notation of the running times. To be more precise, we can see the algorithm in [26] as a *parameterized algorithm* with running time $f(k) \cdot n^3$ for some function f. Towards improving f, we should first of all mention that Robertson and Seymour in [26] did not give any specific bound for f, however, they explicitly mentioned that f can be constructed. This function f is given in the - very technical- proof of the celebrated *Unique Linkage Theorem* in [26] and is responsible for an immense parameter dependence in the running time of the algorithm. Hence two directions of research are: simplify parts of the original proof for the general case or focus on specific classes of graphs that may admit proofs and algorithms with better parameter dependence. An important step in the first direction was done by Kawarabayashi and Wollan in [18] who gave a shorter proof of the results in [27, 28] and yielded an upper bound for $f(k)$ that, however, is (at least) *quadruply exponential in* k.

Towards the second direction, for the case where the input graph is planar, i.e., the PLANAR DISJOINT PATHS problem (in short PDPP), after some results in [25, 30] for planar graphs, a big step was achieved in [2, 3] where an algorithm with a better parametric dependence was presented. According to [3] there is a singly-exponential function f such that every vertex that is insulated by the terminals by a collection of $f(k)$ pairwise vertex-disjoint cyclic separators is irrelevant. If the treewidth of G is more than $c \cdot f(k)$ (for some adequate c) then such an irrelevant vertex can be detected in linear time. Therefore PDPP can be solved in $2^{2^{O(k)}} n^2$ steps [3]. Moreover, in [2] it was argued that the application of the irrelevant vertex technique cannot improve this running time to a singly-exponential one.

1.2 About this thesis

In this thesis we deal with the PLANAR DISJOINT PATHS problem. In fact, we improve the algorithm of [3] to a *linear* one with the same parametric dependence, i.e., it runs in $2^{2^{O(k)}} n$ steps. First, we notice that even on planar graphs, an $O(n^{2-\epsilon})$ step algorithm seems unlikely if we insist on detecting and removing irrelevant vertices one at a time. Indeed, finding an irrelevant vertex in isolation requires a linear number of steps and in the worst case there is a linear number of such vertices to discard. As a consequence, this approach is not liable to provide anything better than a quadratic algorithm. In our work, we overcome this bottleneck by designing a *linear time* algorithm for PDPP, for each fixed k. In particular, we show how to detect in linear time, a set S of vertices of G that can simultaneously be discarded from G so that the remaining graph G' has bounded treewidth. In other words, given an instance $(G, \mathcal{T})$ of PDPP, the algorithm outputs an induced subgraph G' of G containing all the terminals in $\mathcal{T}$ such that $(G, \mathcal{T})$ and $(G', \mathcal{T})$ are equivalent instances. As G' has bounded treewidth, the problem can then be solved in linear time, by dynamic programing.

Our technique. As we already mentioned, the idea is to simultaneously remove all vertices of a suitable set S from a planar embedding of G so that the remaining graph has treewidth $2^{O(k)}$ – we call such an S an *irrelevant set*. We work on the radial graph of G, that is the plane bipartite graph R_G whose vertices are the vertices and the faces of G and where edges correspond to incidences between vertices and faces. For each pair of terminals, we compute a shortest path joining them in R_G. Consider the vertex sets $\mathcal{R} = \{R_1, \ldots, R_m\}$ of the connected components of the subgraph of R_G that is induced by the vertices of these paths and their neighbors in R_G (clearly $m \leq k$). Our main result is that the set S of all vertices of G that are within distance at least $g(k) := 2 \cdot k \cdot f(k)$ from all the vertices in $R = R_1 \cup \cdots \cup R_m$ in R_G is an irrelevant set (where f is the aforementioned singly-exponential function of [3]). Given this, the desired bound on the treewidth follows by a theorem of [6] asserting that, for such an S, $G \setminus S$ has treewidth that is linear in $g(k)$.

The main combinatorial structure, used to prove the irrelevance of S, is a collection $\mathcal{C}$ of pairwise non-crossing cyclic separators of G around the vertices of R, introduced in Chapter 3. The definition of $\mathcal{C}$ is derived from a decomposition of G with respect to the radial distances from the terminals. We next consider some suitable partition $\{\mathcal{R}_1, \ldots, \mathcal{R}_q\}$ of $\mathcal{R}$ (see Lemma 4.1.3) and a corresponding partition $\{S_1, \ldots, S_q\}$

of S such that, for each $i \in \{1, \ldots, q\}$ and each vertex in S_i we can choose from $\mathcal{C}$ a collection of $f(k)$ *pairwise vertex-disjoint cyclic separators* isolating the sets in $\mathcal{R}_i$ from the vertices in S_i. This allows us to apply the main result in [3], for each $i \in \{1, \ldots, q\}$, as follows: if $(G, \mathcal{T})$ is a yes instance of PDPP, then the sub-instance $(G, \mathcal{T}_i)$ induced by the pairs of the terminals of $\mathcal{T}$ that belong in the sets of $\mathcal{R}_i$ has an equivalent solution that avoids S_i. By the way $\mathcal{C}$ is constructed, we can prove that this new solution avoids the entire S (not just S_i). Then, taking the union of all these partial solutions, for each $i \in \{1, \ldots, q\}$, we can build an equivalent solution that avoids S, as required.

The results in this thesis represent joint work with Petr A. Golovach, Stavros G. Kolliopoulos, and Dimitrios M. Thilikos.

CHAPTER 2

PRELIMINARIES

We use $\mathbb{N}$ to denote the set of all nonnegative integers. Given a positive integer k we denote $[k] = \{1, \ldots, k\}$. If $\mathcal{S}$ is a collection of objects where the operation $\cup$ is defined, then we denote $\pmb{\cup}\mathcal{S} = \bigcup_{X \in \mathcal{S}} X$.

2.1 Graphs

All graphs in this thesis are finite and, unless otherwise is mentioned, do not have multiple edges. Also we will make use of both directed and undirected graphs. Given a graph G, we denote its vertex and edge set by $V(G)$ and $E(G)$ respectively. Given some $S \subseteq V(G)$, we denote by $G \setminus S$ the graph obtained if we remove from G the vertices in S, along with their incident edges. For $v \in V(G)$, we denote $G \setminus v = G \setminus \{v\}$. We also denote $G[S] = G \setminus (V(G) \setminus S)$ and we call $G[S]$ *the subgraph of* G *induced by* S. If G' is a graph where $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G[V(G')])$ then we say that G' is *a subgraph of* G. We define $N_G(S)$, as the set of all endpoints of edges that are incident to a vertex in S and do not belong in S. Given a vertex $v \in V(G)$ we set $N_G(v) := N_G(\{v\})$. We call $N_G(v)$ *the neighborhood of* v *in* G and the vertices of $N_G(v)$ *the neighbors of* v *in* G.

Connectivities. Given two vertices x and y of G we define their distance in G as the minimum length of a path in G with endpoints x and y and we denote it by $\mathbf{dist}_G(x, y)$. If such a path does not exist then we say that $\mathbf{dist}_G(x, y) = \infty$. We say that G is *connected* if $\forall x, y\ \mathbf{dist}_G(x, y) < \infty$. A *cut-vertex* of G is a vertex $v \in V(G)$ such that $G \setminus v$ is not connected. We say that G is *2-connected* if it does not contain cut-vertices. A *block* of G is a maximal 2-connected subgraph of G. A block of G is a *leaf-block* if it contains at most one cut-vertex.

Planar and Plane graphs. In most of the cases, the graphs considered in this thesis are *plane graphs*, that is graphs embedded in the sphere without crossing edges. Graphs that admit such an embedding are called *planar graphs*. Given a plane graph G, we denote by $F(G)$ the set of its faces. The *dual* G^* of a plane graph G is a plane graph that has one vertex for each face of G and also there is an edge between two vertices of

G^* if and only if the boundaries of their corresponding faces in G share an edge. Also, if $S \subseteq V(G)$ we denote by S^* the faces of G^* that are dual to the vertices of S. If $F \subseteq F(G)$ we define F^* analogously.

Directed graphs. Given a directed graph D we define its *underlying graph* as the undirected graph obtained if we replace every directed edge by an edge and suppress edge multiplicities. Given a vertex v of D we call *in-neighbors* of v all the vertices of D that are tails of edges heading at v and *out-neighbors* of v all the vertices of D that are heads of edges tailing at v.

Cuts. A *cut* (S, T) of G is a partition of $V(G)$ into two subsets S and T. The *cut-set* of a cut (S, T) is the set of edges of G that have one endpoint in S and the other endpoint in T. A minimal non-empty cut-set is a *bond*.

Treewidth. Given a $k \in \mathbb{N}^+$, we say that a graph G is a *k-tree* if G is isomorphic to K_{k+1} or (recursively) there is a vertex v in G where $N_G[\{v\}]$ isomorphic to K_{k+1} and $G \setminus \{v\}$ is a k-tree. The *treewidth* of a G is the minimum k for which G is a subgraph of some k-tree.

2.2 Parameterized problems and algorithms

Problem parameterization is a concept introduced in theoretical computer science as a way (among approximation and randomness) of coping with NP problems. The idea is to treat algorithmic problems as parameterized entities and compute the complexity of the corresponding algorithm by considering the way the parameter affects the running time of the algorithm. *Parameterized Complexity* as an area related to the study of such *parameterized algorithms* and the notion of *tractability* and *efficiency* in this context has gathered significant attention recently. We refer to [4] as an introductory but yet detailed book in Parameterized Complexity. Since here we deal with problems on graphs, we present some classic definitions of parameterized complexity in the form where problem inputs represent graphs.

Let Σ be an alphabet and let Σ^* (the *Kleene star* of Σ) be the set of all finite sequences with elements from Σ.

Formally, a *parameterized problem on graphs* is a subset Π of $\Sigma^* \times \mathbb{N}$ where in each $(I, k) \in \Sigma^* \times \mathbb{N}$, I encodes a combinatorial structure related to one, or more, graphs. We denote by n the maximum size of the graphs encoded in I and insist that $|(I, k)| = O(n)$. We call I the *main part of the input* and we say that k is the *parameter* of the problem.

We say that Π is *fixed parameter tractable* if there exists a function $f : \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm deciding whether $(I, k) \in \Pi$ in $O(f(k) \cdot n^c)$ steps, where c is a constant not depending on the parameter k of the problem. We call such an algorithm an FPT-algorithm. A parameterized problem on graphs belongs to the parameterized class FPT if it can be solved by an FPT-algorithm. In fact, not all parameterized problems belong to the class FPT and the study of parameterized problems has led researchers to define some hierarchies of paramererized complexity classes (as W-hierarchy or A-hierarchy) following the respective work in classical Complexity Theory.

CHAPTER 3

DECOMPOSITIONS OF PLANE GRAPHS

In this chapter we deal with decompositions of graphs and our aim is to define a decomposition of a plane graph based on the radial distances of its vertices and faces from the terminals. Next we prove a series of properties of such decompositions.

3.1 Layered decompositions

Leveled DAG. A directed graph $Q = (V, E)$ is a *Leveled Directed Acyclic Graph*, in short LDAG, when the following conditions are satisfied:

- the underlying graph of Q is acyclic
- there exists a partition $\{L_0, \ldots, L_\ell\}$ of V such that
  - for every edge $xy \in E$, if $y \in L_i$, then $x \in L_{i-1}$ for some $i \in [\ell]$.
  - for every $i \in [\ell]$ and $x \in L_i$ there is an edge $yx \in E$ such that $y \in L_{i-1}$.

We call the sets $L_0, \ldots, L_\ell$ *levels* of Q and we call ℓ *the depth* of Q. If $v \in L_i$ for some odd/even i, then we say that v is an *odd/even vertex* of Q. Notice that the vertices in L_0 are the vertices of Q without in-neighbors. We refer to these vertices as the *root* vertices of Q. If Q has only one root then we call it *single-rooted* and we denote the root of Q by r_Q. Given a $i \in [0, \ell]$, we set $L_{\leq i} = \bigcup_{j \in [0,i]} L_i$ and $L_{\geq i} = \bigcup_{j \in [i,\ell]} L_i$.

LDAG decomposition. Let G be a connected graph. We say that $\mathcal{R} = \{R_1, \ldots, R_m\}$, $m \geq 1$ is a *root collection* of G if it consists of pairwise disjoint connected subsets of $V(G)$. Given an $i \in \mathbb{N}$, we define D_i as follows: $D_0 = \bigcup_{j \in [m]} R_j$ and, for $i \geq 1$, we set $D_i = N_G(D_{i-1}) \setminus \bigcup_{j \in [i-1]} D_j$.

We define the *eccentricity* of $\mathcal{R}$ as the maximum i for which D_i is non-empty and we always use ℓ to denote the eccentricity of $\mathcal{R}$. We also define $D_{\leq i} = \bigcup_{j \in [0,i]} D_j$ and $D_{\geq i} = \bigcup_{j \in [i,\ell]} D_j$. We define an equivalence relation between vertices as follows: given $x, y \in V(G)$, we say that $x \sim_{\mathcal{R}} y$ if the following hold:

- $\exists i \in [0, \ell]$ such that $x, y \in D_i$,
- there is an (x, y) path in $G[D_{\leq i}]$, and
- there is an (x, y) path in $G[D_{\geq i}]$.

Notice that $\sim_{\mathcal{R}}$ is an equivalence relation that partitions $V(G)$ into equivalence classes. Also, the vertices that belong in different D_i's cannot be equivalent. Moreover, for every $i \in [0, \ell]$, D_i is the union of, say d_i equivalence classes of $\sim_{\mathcal{R}}$, which we denote by $X_{i,j}, j \in [d_i]$. Clearly $\{X_{i,j} \mid j \in [d_i], i \in [0, \ell]\}$ is a refinement of $\{D_i \mid i \in [0, \ell]\}$.

We build a directed graph $Q := Q_{\mathcal{R}}(G)$ so that its vertex set is $L_0 \cup L_1 \cup \cdots \cup L_\ell$ where $L_i = \{(i, j) \mid j \in [d_i]\}$ and an edge $((i, j), (i', j'))$ exists if $i' = i + 1$ and there exists an edge of G with one endpoint in $X_{i,j}$ and the other in $X_{i',j'}$. We say that a vertex $x = (i, j)$ is a *fusion* vertex of Q if $\deg_Q^{\text{in}}(x) > 1$. Notice that $Q_{\mathcal{R}}(G)$ is an LDAG. We refer to the pair $(\mathcal{X}, Q)$ where $\mathcal{X} = \{X_{i,j} \mid (i, j) \in V(Q)\}$ as the *LDAG-decomposition of G with respect to the root collection $\mathcal{R}$*. We also refer to $D_0, \ldots, D_\ell$ as the *layers* of $(\mathcal{X}, Q)$. If Q is single-rooted then we simply denote the root of Q by r_Q.

Recall that Q is connected and $Q[L_0]$ has m connected components (the roots of Q). Moreover, each fusion vertex, when it appears, decreases the number of connected components by at least one. This implies that Q has at most $m - 1$ fusion vertices. This combined with the pigeonhole principle, yields the following.

Observation 3.1.1. Let s be a positive even integer, G be a graph, $(\mathcal{X}, Q)$ be the LDAG-decomposition of G with respect to some root collection $\mathcal{R} = \{R_1, \ldots, R_m\}$, and let $\{L_0, \ldots, L_\ell\}$ be the levels of Q. If $\ell > s \cdot m$, then there is a non-negative even integer $p \leq s \cdot (m - 1)$ such that none of the vertices in the levels $L_{p+1}, \ldots, L_{p+s}$ is a fusion vertex.

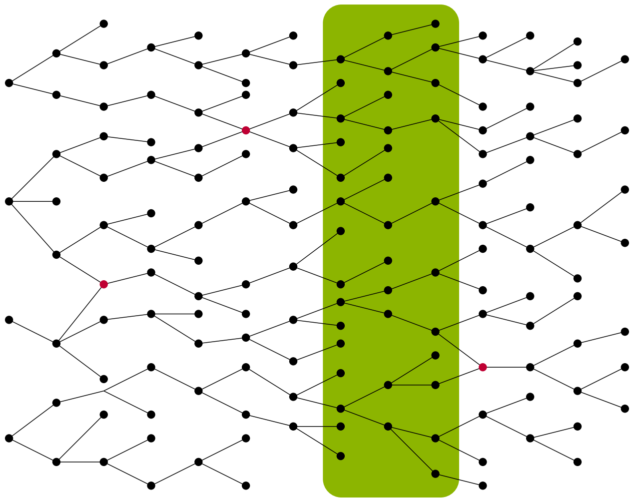

Figure 3.1: An example of a graph Q of the LDAG decomposition $(\mathcal{X}, Q)$ of some graph G with respect to some root collection $\mathcal{R}$, where $|\mathcal{R}| = 4$. Q has 3 fusion vertices (depicted in red) and depth 13. By setting $s = 3$, then Observation 3.1.1 holds (i.e. for $p = 6$, observe that none of the vertices in L_7, L_8, L_9 (depicted as a green "window") is a fusion vertex).

3.2 Radial graphs and strongly connected sets

Plane graphs and strongly-connected sets Let G be plane graph and let F be a subset of its faces. We say that F is *strongly-connected* in G if for every two faces f_1, f_2 in F there is a $V(G)$-avoiding arc (that is a subset of the sphere that is homeomorphic with the closed interval $[0, 1]$) starting from a point in f_1 and finishing to a point in f_2 and not containing any point from a face outside F. Observe that $F \subseteq F(G)$ is strongly-connected in G iff $G^*[F^*]$ is connected.

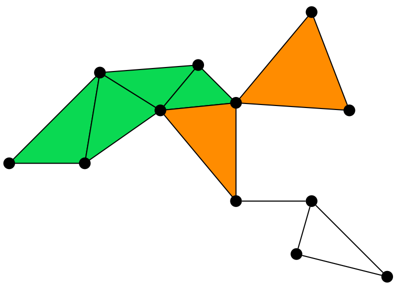

Figure 3.2: An example of a plane graph G, a strongly-connected subset of its faces (depicted in green), and a subset of its faces that is not strongly-connected (depicted in orange).

The definition above easily implies the following two results.

Observation 3.2.1. If G is a 2-connected plane graph, then $F(G)$ is strongly-connected in G.

Observation 3.2.2. If G is a plane graph and $F_0, F_1, \ldots, F_r$ are pairwise-disjoint subsets of $F(G)$ such that $F_0 \cup F_i$ is strongly-connected in G, then $\bigcup_{i \in [0,r]} F_i$ is strongly-connected in G.

We now prove the following lemma concerning the strong connectivity of two sets of faces that correspond to the bipartition (i.e. partition in two parts) of the faces that are incident to a vertex of a 2-connected plane graph.

Lemma 3.2.3. *Let G be a 2-connected plane graph, let $v \in V(G)$, and let $\mathcal{F}$ be the faces of G that are incident to v. If $\mathcal{F}'$ is a subset of $\mathcal{F}$ where $\bigcup \mathcal{F}'$ is strongly-connected, then $\bigcup(\mathcal{F} \setminus \mathcal{F}')$ is strongly-connected.*

Proof. Let $\{u_1, \ldots, u_m\} = N_G(v)$, for some $m \geq 2$. Let also $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be the faces of G incident to v, following the ordering of the neighbors of v, i.e., we assume that f_m is the face of G that contains the edges vu_m, vu_1 in its boundary and for $i \in [m-1]$, f_i is the face of G that contains vu_i, vu_{i+1} in its boundary. Observe that since G is 2-connected, then $f_i \neq f_j, \forall i, j \in [m]$. Let $I \subseteq [m]$ be the indices of the faces in $\mathcal{F}'$. Since $\bigcup \mathcal{F}'$ is strongly-connected, the indices in I are consecutive in the cyclic ordering $\{1, \ldots, m, 1\}$. This implies that the indices of $[m] \setminus I$ are also consecutive in the cyclic ordering $\{1, \ldots, m, 1\}$, therefore $\bigcup(\mathcal{F} \setminus \mathcal{F}')$ is strongly-connected. □

Now we present the notion of the *radial graph*, a combinatorial object that is crucial to the construction of our decomposition.

Radial graphs. Given a plane graph G, we define the *radial graph* of G as the bipartite plane graph $R_G = (V(G) \cup F(G), E)$ whose edge set E is defined as follows: for every $f \in F(G)$ we consider the closed walk of G defined by the boundary of f and we make f adjacent to all the vertices in this walk (we permit multiple edges as a vertex can appear many times in the walk). Notice that if G is 2-connected then the dual G^* is a loop-less plane graph.

We say that a vertex v in R_G is a v-*vertex* of R_G if $v \in V(G)$ while if $v \in F(G)$, it is an f-*vertex* of G.

Let S be a subset of $V(R_G)$. We say that S is *normal* in R_G if $N_{R_G}(S) \subseteq V(G)$. Also, we extend the notion of strong connectivity on any normal set S of R_G by saying that S is strongly-connected in R_G if $F(G) \cap S$ is strongly-connected in G. Notice that if S is strongly-connected in R_G, then it is also connected in R_G.

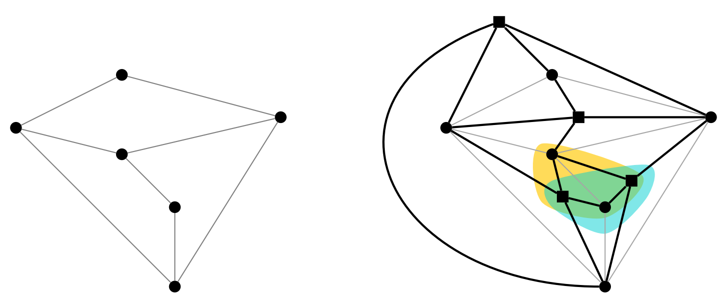

Figure 3.3: On the left, an example of a graph G. On the right, its radial graph R_G, a normal set (depicted in blue), and a set that is not normal (depicted in yellow).

A direct consequence of Lemma 3.2.3:

Corollary 3.2.4. *Let G be a 2-connected plane graph and let $v \in V(G)$. Then $N_{R_G}(v)$ is strongly-connected in R_G.*

We now use Corollary 3.2.4 to prove that a normal set must also be strongly-connected.

Lemma 3.2.5. *Let G be a 2-connected plane graph. If Z is a connected normal subset of $V(R_G)$, then Z is strongly-connected.*

Proof. Let $f, f' \in F(G) \cap Z$. It is enough to prove that there is a path connecting f and f' in $G^*[Z^*]$. Since Z is connected, then there exists a path P in $R_G[Z]$ whose vertices are $f_0 = f, v_1, f_1, \ldots, v_{m-1}, f_{m-1}, v_m, f_m = f'$, starting from f and finishing at f'. Since Z is a normal subset of $V(R_G)$, then for every $i \in [m]$, $N_{R_G}[v_i] \subseteq V(R_G[Z])$. Thus, by Corollary 3.2.4, there exists a path in $G^*[Z^*]$ between f_{j-1} and f_j for every $j \in [m]$ and therefore also a path connecting f and f' in $G^*[Z^*]$. □

Before concluding this section, we present an example. In Figure 3.4, we show a 2-connected graph G, a connected normal subset of $V(R_G)$, the LDAG decomposition $(\mathcal{X}, Q)$ of R_G with respect to $\{S\}$, and the levels of Q, shown on G. In Figure 3.5, we show the graph Q of the LDAG decomposition $(\mathcal{X}, Q)$ of R_G with respect to $\{S\}$.

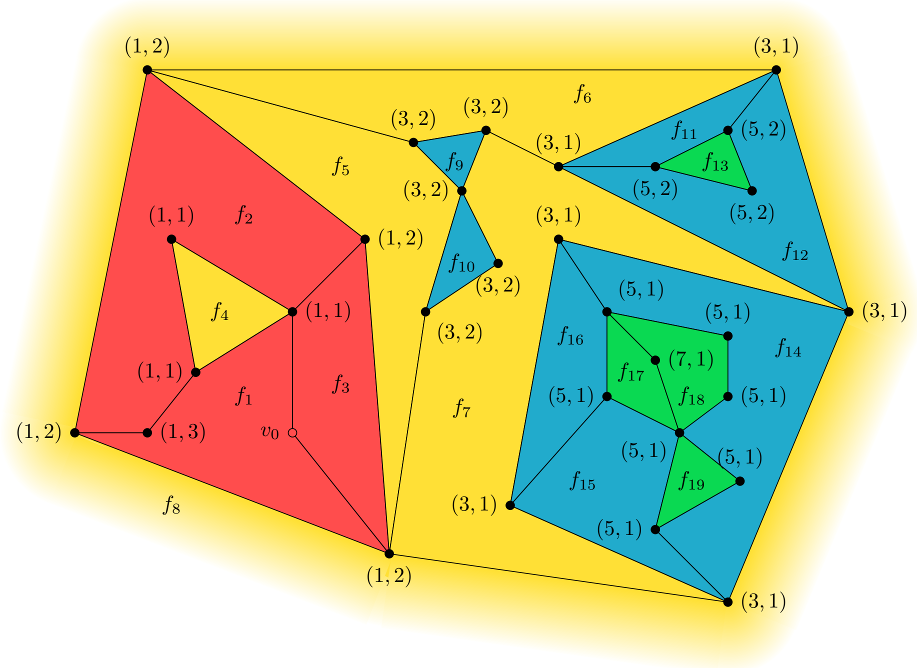

Figure 3.4: A 2-connected graph G , a connected normal subset $S = \{v_0, f_1, f_2, f_3\}$ of $V(R_G)$ (the vertex v_0 together with the faces depicted in red), and the corresponding LDAG decomposition $(\mathcal{X}, Q)$ of R_G with respect to $\{S\}$. The indices in the vertices correspond to the sets $X_{i,j}$ of $\mathcal{X}$ while same-colored faces are in the same (even) layer of $(\mathcal{X}, Q)$.

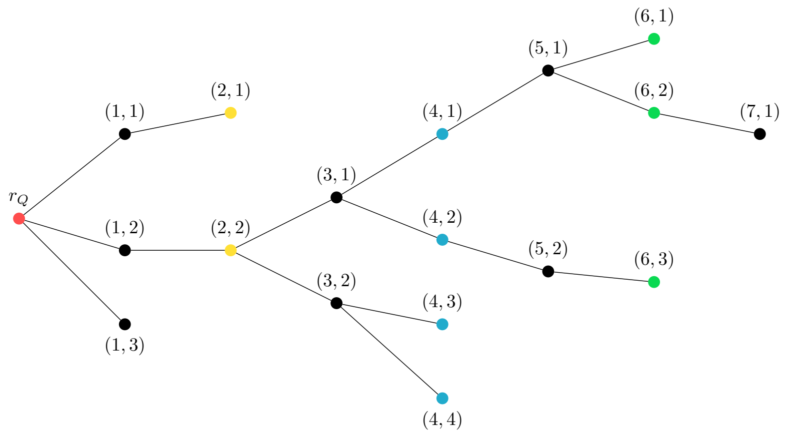

Figure 3.5: The graph Q of the LDAG decomposition $(\mathcal{X}, Q)$ in Figure 3.4. The vertex r_Q corresponds to S, while $(2,1) = \{f_4\}$, $(2,2) = \{f_5, f_6, f_7, f_8\}$, $(4,1) = \{f_{14}, f_{15}, f_{16}\}$, $(4,2) = \{f_{11}, f_{12}\}$, $(4,3) = \{f_9\}$, $(4,4) = \{f_{10}\}$, $(6,1) = \{f_{19}\}$, and $(6,2) = \{f_{17}, f_{18}\}$. The coloring of the even vertices of Q follows the coloring in Figure 3.4.

3.3 Propagating strong connectivity

The purpose of this section is to show that, for every 2-connected plane graph G, given a single-rooted LDAG-decomposition of the radial graph of G, each edge of the underlying DAG connecting an odd vertex with an even vertex corresponds to a partition of the faces of G into two strongly connected sets and therefore to a cyclic separator of G.

Suffixes and prefixes. Let $(\mathcal{X}, Q)$ be a single-rooted LDAG-decomposition of a 2-connected plane graph. Notice that Q does not have fusion vertices.

Let $e \in E(Q)$. Notice that $Q \setminus e$ has two connected components. We say that the connected component of Q that contains the root of Q is the *Q-prefix of e* while the other component is the *Q-suffix of e*. Given a vertex $v \in V(Q)$, we define the *Q-prefix of v* as the union of $\{v\}$ with the Q-prefix of the (unique due to the absence of fusion vertices) edge of Q pointing to v, while we define the *Q-suffix of v* as the union of $\{v\}$ with the Q-suffix of every edge of Q starting from v. We also define the $\mathcal{X}$-prefix/suffix of e (resp. v) as the union of all X_u where u is in the Q-prefix/suffix of e (resp. v).

We prove the next lemma:

Lemma 3.3.1. *Let G be a 2-connected plane graph and let S be a connected normal subset of $V(R_G)$, and let $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\{S\}$. Let x be an odd vertex of Q and $\mathcal{A}$ be the vertex sets of the connected components of $V(R_G) \setminus X_x$. Suppose also that all sets in $\mathcal{A}$ are strongly-connected. Then for every $B \in \mathcal{A}$ where $S \cap B = \emptyset$, the union of $\bigcup(\mathcal{A} \setminus \{B\})$ and X_x is also strongly-connected in R_G.*

Proof. Let $A_S \in \mathcal{A}$ be the (unique) strongly-connected set of R_G that contains S. Consider a strongly-connected set $B \in \mathcal{A} \setminus \{A_S\}$. Let $v \in X_x$ and denote $\overline{B} =$

$\bigcup(\mathcal{A} \setminus \{B\})$. Let also $\mathcal{F}$ be the set of faces of G that are incident to v and let $\mathcal{F}_B$ be the set of faces of G corresponding to f-vertices of B.

Suppose to the contrary that $\overline{B} \cup X_x$ is not strongly-connected in R_G. Let $C_1, \dots, C_r$, $r \geq 2$ be the connected components of $G^* \setminus (\mathcal{F}_B)^*$. Let $\mathcal{F}_i$ be the faces of C_i^* that are incident to v. Let also $\mathcal{F}_0$ be the faces of B that are incident to v. Clearly $\{\mathcal{F}_0, \mathcal{F}_1, \dots, \mathcal{F}_r\}$ is a partition of $\mathcal{F}$. Notice that for every $i \in [r]$ there is some neighbor u_i of v that is incident both to a face in $\mathcal{F}_0$ and to a face in $\mathcal{F}_i$. Also, observe that for every $i \in [r]$, $u_i \in B \cup X_x$ and also $u_i \in \overline{B} \cup X_x$, which implies that $u_i \in X_x$. This, in turn implies that for every $i \in [r]$ there is a face f_i of A_S such that u_i is incident to f_i and $f_i \in \mathcal{F}_i$. We arrive at a contradiction to the fact that A_S is strongly-connected. □

The result of the next lemma is a key step towards building an induction so as to prove Lemma 3.3.3.

Lemma 3.3.2. *Let G be a 2-connected plane graph and let S be a connected normal subset of $V(R_G)$. Let also $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\{S\}$. Then the following hold:*

1. *For every $e = xy \in E(Q)$, where x is an odd vertex of Q an y is an even vertex of Q, the $\mathcal{X}$-suffix of e is strongly-connected in R_G.*

2. *For every $e = xy \in E(Q)$, where x is an even vertex of Q an y is an odd vertex of Q, it holds that if the $\mathcal{X}$-prefix of x is strongly-connected in R_G, then the $\mathcal{X}$-prefix of e is strongly-connected in R_G.*

3. *For every pair of edges $e, e' \in E(Q)$ such that $e = xy$, $e' = yz$, where x, z are even vertices of Q and y is an odd vertex of Q, it holds that if the $\mathcal{X}$-prefix of e is strongly-connected in R_G, then the $\mathcal{X}$-prefix of e' is strongly-connected in R_G.*

4. *For every $e = xy \in E(Q)$ where x is an odd vertex of Q and y is an even vertex of Q, it holds that if the $\mathcal{X}$-prefix of e is strongly-connected in R_G, then the $\mathcal{X}$-prefix of y is strongly-connected in R_G.*

Proof. (1) Let Z be the $\mathcal{X}$-suffix of e. Observe that Z is connected and since x is an odd vertex, then Z is also a normal subset of $V(R_G)$. The desired result follows by Lemma 3.2.5.

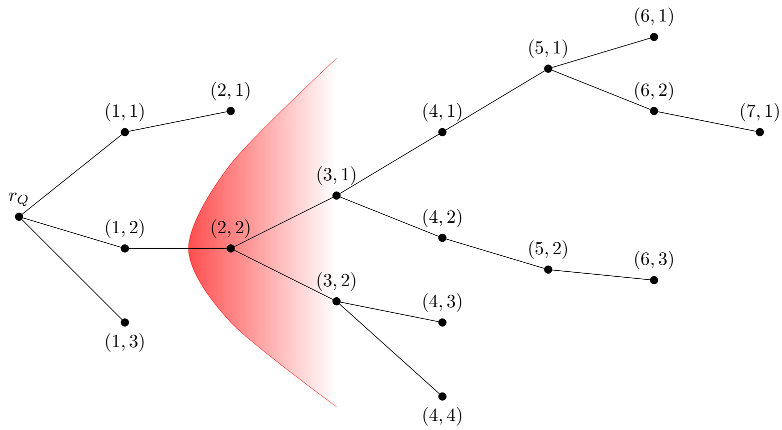

Figure 3.6: The $\mathcal{X}$-suffix of $\big((1,2),(2,2)\big)$ in the example of Figure 3.4.

(2) Consider an edge $e = xy \in E(Q)$ where x is an even vertex of Q an y is an odd vertex of Q, such that the $\mathcal{X}$-prefix of x is strongly-connected in R_G. Let $u_1, \ldots, u_m$ be all out-neighbors of x in Q, except y. Also, let $E_i = \{e \in E(Q) \mid e = u_i w \text{ for some } w \in V(Q)\}$, $i \in [m]$.

By (1), we have that for every $i \in [m]$ and for every edge $e' \in E_i$, the $\mathcal{X}$-suffix of e' is strongly-connected in R_G. For every $i \in [m]$, let A_i be the union of the $\mathcal{X}$-suffix, we call it $Z_{e'}$, of every edge $e' \in E_i$ with the $\mathcal{X}$-prefix, we call it Z_x, of x.

Claim: For every $i \in [m]$, A_i is strongly-connected in R_G.

Proof of claim: We fix an f-vertex f of Z_x. Let F_i be the set containing the faces in Z_x and the faces in $\bigcup_{e' \in E_i} Z_{e'}$. It is enough to prove that for every f-vertex f' of $Z_{e'}$ for some $e' \in E_i$, there exists a path connecting f with f' in G^* consisting of faces in F_i. Notice that there is a vertex $v \in X_{u_i}$ that is incident to a face g of Z_x and a face g' of $Z_{e'}$. Also, since Z_x (resp. $Z_{e'}$) is strongly-connected, then there exists a path P_1 (resp. P_2) in G^* that from f (resp. f') to g (resp. g') consisting of faces in F_i. Notice that the faces, call them F, of G that are incident to v are also faces of F_i. Therefore the set F is partitioned into two sets, one consisting of faces of Z_x and the other consisting of faces in $\bigcup_{e' \in E_i} Z_{e'}$. This implies the existence of a path $P^\bullet$ in G^* from g to g' consisting of faces in F_i. By now joining the paths $P_1, P^\bullet, P_2$ we construct a path from f to f' as claimed.

Now (2) follows by the above claim and applying Observation 3.2.2, on the set of faces of Z_x and the sets of faces in $\bigcup_{e' \in E_i} Z_{e'}$, $i \in [m]$.

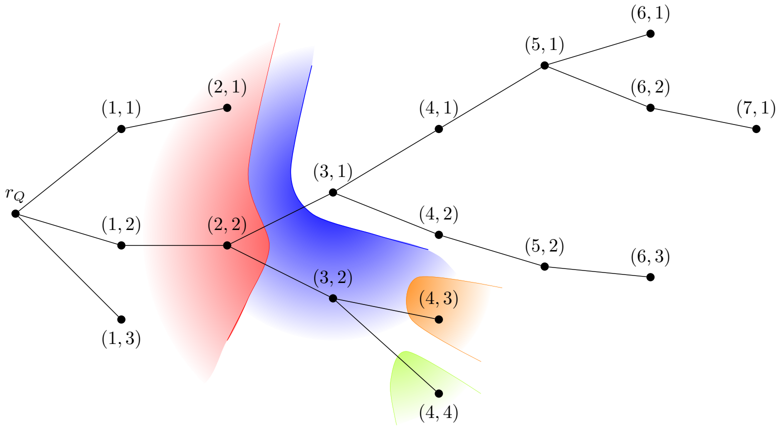

Figure 3.7: The $\mathcal{X}$-prefix of $(2,2)$ (depicted in red) and the $\mathcal{X}$-prefix of $\big((2,2),(3,1)\big)$ (depicted in blue) in the example of Figure 3.4. Also, in this example $u_1 := (3,2)$, $E_1 = \{e', e''\} := \{\big((3,2),(4,3)\big), \big((3,2),(4,4)\big)\}$, the $\mathcal{X}$-suffix of e' (depicted in orange), the $\mathcal{X}$-suffix of e'' (depicted in green), and the set A_1 that is the union of the red, the orange, and the green area.

(3) Consider some edges $e, e' \in E(Q)$ such that $e = xy$, $e' = yz$, where x, z are even vertices of Q, y is an odd vertex of Q and the $\mathcal{X}$-prefix, we call it A, of e is strongly-connected in R_G. Let $\{u_1, \ldots, u_m\}$ be the set of all out-neighbors of y, except z. By (1), we have that for every $i \in [m]$, the $\mathcal{X}$-suffix B_i of yu_i is strongly-connected in R_G, and the same holds for the $\mathcal{X}$-suffix of yz. Observe that the collection $\mathcal{U} = \{A, B_1, \ldots, B_m\}$ together with the $\mathcal{X}$-suffix of yz, form a partition of $V(R_G) \setminus \{X_y\}$. Therefore, by Lemma 3.3.1, the $\mathcal{X}$-prefix of e', that is the union of $\mathcal{U}$ with X_y, is strongly-connected in R_G.

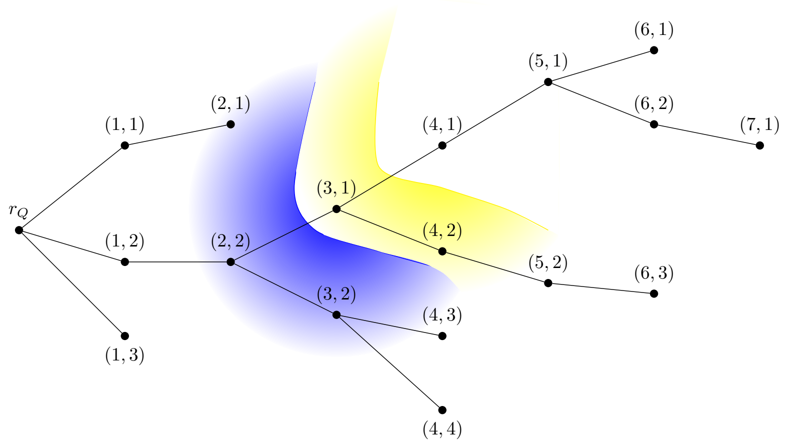

Figure 3.8: The $\mathcal{X}$-prefix of $\big((2,2),(3,1)\big)$ (depicted in blue) and the $\mathcal{X}$-prefix of $\big((3,1),(4,1)\big)$ (depicted in yellow) in the example of Figure 3.4.

(4) Consider an edge $e = xy \in E(Q)$ where x is an odd vertex of Q and y is an even vertex of Q, such that the $\mathcal{X}$-prefix A of e is strongly-connected in R_G. Let B be the $\mathcal{X}$-prefix of y and let $f \in B \setminus A$. It is enough to prove that there is a path in G^* from f to some face in A consisting of faces in B.

Let v be a vertex of X_x such that v is incident to both f and some face in A. Notice that the faces, call them F, of G that are incident to v are also faces in B. Therefore the set F is partitioned into two sets, one consisting of faces in A and the other consisting of faces in $B \setminus A$. This implies the existence of a path P in G^* from f to some face in A consisting of faces in F, as required. □

Now, we have all necessary tools to prove the next lemma.

Lemma 3.3.3. *Let G be a 2-connected plane graph and let S be a strongly-connected normal subset of $V(R_G)$. Let also $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\{S\}$. Then for every $e = xy \in E(Q)$ where x is an odd vertex of Q and y is an even vertex of Q, both the $\mathcal{X}$-prefix and the $\mathcal{X}$-suffix of e are strongly-connected in R_G.*

Proof. By Lemma 3.3.2(1), for every edge $xy \in E(Q)$ where x is an odd vertex of Q and y is an even vertex of Q, it holds that the $\mathcal{X}$-suffix of xy is strongly-connected in R_G. Suppose towards a contradiction that there exists an edge $xy \in E(Q)$ where x is an odd vertex of Q and y is an even vertex of Q, such that the $\mathcal{X}$-prefix of xy is not strongly-connected in R_G. As S is strongly-connected in R_G, we have that x is not the (unique) root r_Q of the LDAG-decomposition $(\mathcal{X}, Q)$. We pick $e = xy$ so that x is at the minimum possible distance from r_Q. Let $e' = zx$ be the edge of Q pointing to x and keep in mind that z is an even vertex of Q. Also, assume that $z \neq r_q$, for if otherwise, by Lemma 3.3.1, the $\mathcal{X}$-prefix of e, that is the union of S with the $\mathcal{X}$-suffix of every edge of Q starting from r_q, other than e', is strongly-connected.

Therefore, there exists an edge $e'' = wz \in E(Q)$, where w is an odd vertex of Q. Observe that e'' is the unique edge of Q pointing to z, due to the absence of fusion vertices. By the minimality of e, it holds that the $\mathcal{X}$-prefix of e'' is strongly-connected in R_G. Therefore, by applying successively Lemma 3.3.2(4), Lemma 3.3.2(2) and Lemma 3.3.2(3), we obtain that the $\mathcal{X}$-prefix of e is also strongly-connected in R_G, a contradiction to our initial assumption. □

The existence of the cyclic separators claimed in the beginning of this section is proved in the next lemma.

Lemma 3.3.4. *Let G be a 2-connected plane graph and let S be a strongly-connected normal subset of $V(R_G)$. Let also $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\{S\}$. Then for every $e = xy \in E(Q)$ where x is an odd vertex of Q and y is an even vertex of Q, there is a cycle in G bounding a closed disk D such that*

- *each vertex or face of G that belongs in the $\mathcal{X}$-prefix of e is a subset of D*
- *each vertex or face of G that belongs in the $\mathcal{X}$-suffix of e does not intersect D, and*
- *$V(C) \subseteq X_x$.*

Proof. Consider an edge $e = xy \in E(Q)$ where x is an odd vertex of Q and y is an even vertex of Q. By Lemma 3.3.3, both the $\mathcal{X}$-prefix and the $\mathcal{X}$-suffix of e are strongly-connected in R_G. Let $\mathcal{F}_{\text{pre}}, \mathcal{F}_{\text{suf}}$ be the sets of all faces of G that are in the $\mathcal{X}$-prefix and the $\mathcal{X}$-suffix of e, respectively. Since both the $\mathcal{X}$-prefix and the $\mathcal{X}$-suffix of e are strongly-connected in R_G, then both $\mathcal{F}_{\text{pre}}, \mathcal{F}_{\text{suf}}$ are strongly-connected in G. Notice that $\{\mathcal{F}_{\text{pre}}, \mathcal{F}_{\text{suf}}\}$ is a partition of the faces of G. Therefore, $(\mathcal{F}^*_{\text{pre}}, \mathcal{F}^*_{\text{suf}})$ is a cut of G^* and the corresponding cut-set is a bond.

Let Z be the set of faces of G^* whose boundary intersects the cut-set corresponding to the cut $(\mathcal{F}^*_{\text{pre}}, \mathcal{F}^*_{\text{suf}})$ of G^*. Since $\mathcal{F}_{\text{pre}}$ (resp. $\mathcal{F}_{\text{suf}}$) is strongly-connected in G, then $G^*[\mathcal{F}^*_{\text{pre}}]$ (resp. $G^*[\mathcal{F}^*_{\text{suf}}]$) is connected. Therefore, in G, the vertices Z^* induce a cycle C which separates $\mathcal{F}_{\text{pre}}$ and $\mathcal{F}_{\text{suf}}$ in G.

Also, observe that every face in Z is incident to vertices in both $\mathcal{F}^*_{\text{pre}}, \mathcal{F}^*_{\text{suf}}$ and therefore for every vertex $u \in V(C)$ there exist $f \in \mathcal{F}_{\text{pre}}, f' \in \mathcal{F}_{\text{suf}}$ such that u is incident to both f, f'. Moreover, since every vertex of G that is incident to some face in $\mathcal{F}_{\text{pre}}$ and some face in $\mathcal{F}_{\text{suf}}$ is a vertex in X_x, then $u \in X_x$. Thus, $V(C) \subseteq X_x$.

So, C is bounding a closed disk D such that the set of vertices of R_G that belong in the $\mathcal{X}$-prefix of e is a subset of D and the set of vertices of R_G that belong in the $\mathcal{X}$-suffix of e does not intersect D. □

3.4 Finding nested cycles

In the previous section we proved that if G is a 2-connected plane graph and $(\mathcal{X}, Q)$ is the LDAG-decomposition of its radial graph, then each edge of the underlying DAG corresponds to a partition of the faces of G into two strongly connected sets and therefore to a cyclic separator of G. Accordingly, we now show that each path of this DAG corresponds to a collection of nested cyclic separators.

Nested cycles. Let G be a plane graph and let $\mathcal{C} = \{C_1, \ldots, C_r\}, r \geq 2$ be a sequence of cycles in G. We call $\mathcal{C}$ *nested*, if they are pairwise disjoint and, in case $r \geq 3$, the dual of their union contains only one face bounded by more than 2 vertices. For each $i \in [r]$, we define the *disk* of C_i as the closed disk bounded by C_i that contains $C_1, \ldots, C_i$ and does not contain $C_{i+1}, \ldots, C_r$. We say that a vertex set $S \subseteq V(G)$ is *inside* C_i if each of its vertices belongs in its disk but not in C_i. Also, S is *outside* C_i if it does not intersect its disk.

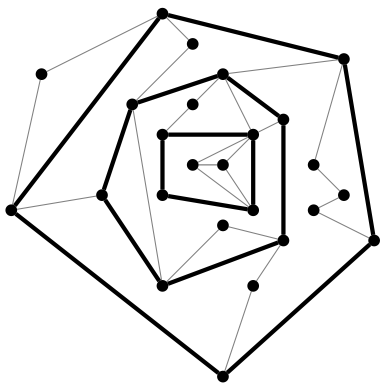

Figure 3.9: An example of a plane graph G and a sequence $\mathcal{C}$ of 3 nested cycles in G.

Lemma 3.4.1. *Let $s \geq 2$ be a integer, G be a 2-connected plane graph and let S be a strongly-connected normal subset of $V(R_G)$, $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\{S\}$, and P be a path of length $2s-1$ in Q whose vertices (following the ordering of the path) are $v_1, f_1, \ldots, v_s, f_s$, starting from an odd vertex of Q and finishing to an even vertex of Q. Then G contains a sequence $C_1, \ldots, C_s$ of nested cycles such that for every $i \in [s]$ S is inside C_i and the set of all* v*-vertices of the $\mathcal{X}$-suffix of f_s is outside C_s.*

Proof. Due to Lemma 3.3.4, for every $v_i f_i$, $i \in [s]$, there is a cycle C_i in G bounding a closed disk D_i such that:

1. each vertex or face of G that belongs in the $\mathcal{X}$-prefix of $v_i f_i$ is a subset of D_i
2. each vertex or face of G that belongs in the $\mathcal{X}$-suffix of $v_i f_i$ does not intersect D_i, and
3. $V(C_i) \subseteq X_{v_i}$.

Notice that (3) implies that the cycles $C_1, \ldots, C_s$ are pairwise disjoint. Also, observe that for every pair of edges $v_i f_i, v_j f_j$ such that $i < j,\ i, j \in [s]$, the $\mathcal{X}$-prefix of $v_i f_i$ is a subset of the $\mathcal{X}$-prefix of $v_j f_j$ and the $\mathcal{X}$-suffix of $v_j f_j$ is a subset of the $\mathcal{X}$-suffix of $v_i f_i$. Therefore, $C_1, \ldots, C_s$ is a sequence of nested cycles.

Furthermore, since for every $i \in [s]$ the $\mathcal{X}$-prefix of $v_i f_i$ contains $X_{v_1}, \ldots, X_{v_i}$ and S, while the $\mathcal{X}$-suffix of $v_i f_i$ contains $X_{v_{i+1}}, \ldots, X_{v_s}$, then, by (1) and (2), D_i is the disk of C_i and for every $i \in [s]$, S is inside C_i and the set of all v-vertices of the $\mathcal{X}$-suffix of f_s is outside C_s. ☐

CHAPTER 4

EQUIVALENT LINKAGES

We now have all the necessary combinatorial tools for finding an equivalent instance of the PDPP that has bounded treewidth. Our next step is to combine the results of the previous section with the main result of [3] in order to rearrange the paths of a solution to the PDPP. In fact we will repeatedly apply [3] along all the collections of nested cycles corresponding to each path of an LDAG-decomposition of R_G. This enables us to confine the solution in a small-radius region around the terminals and makes it possible to bound the treewidth of the remaining graph by the result of [6].

4.1 Rearranging linkages

Linkages. A *linkage* in a graph G is a non-empty subgraph $\mathcal{L}$ of G whose connected components are all paths. The *paths* of a linkage are its connected components and we denote them by $\mathcal{P}(\mathcal{L})$. The *terminals* of a linkage $\mathcal{L}$, denoted by $T(\mathcal{L})$, are the endpoints of the paths in $\mathcal{P}(\mathcal{L})$, and the *pattern* of $\mathcal{L}$ is the set $\big\{\{s,t\} \mid \mathcal{P}(\mathcal{L})$ contains a path from s to t in $G\big\}$. In the definition of a pattern we permit its elements to be mulit-sets (i.e., $s = t$) as a linkage may have a path of length 0. Two linkages are *equivalent* if they have the same pattern. The *size* of a linkage is the number of its connected components.

Let G be a plane graph and let S_1, S_2 be disjoint subsets of $V(G)$. We define the *layer-distance* between S_1 and S_2, denoted by $\mathbf{ldist}_G(S_1, S_2)$, as the maximum r for which there exists a nested sequence of cycles $\mathcal{C} = \langle C_1, \ldots, C_r\rangle$ where S_1 is a subset of the interior of the disk of C_1 and S_2 is a subset of the exterior of the disk of C_r.

The proof of the next proposition implicitly follows from the main result of [3].

Proposition 4.1.1. *There is a function $f : \mathbb{N} \to \mathbb{N}$ such that if G is a planar graph, $\mathcal{L}$ is a linkage in G of size at most k, R is a subset of $V(G)$ such that $\mathbf{ldist}_G(T(\mathcal{L}), R) \geq f(k)$, then there is a linkage $\mathcal{L}'$ in $G \setminus R$ that is equivalent to $\mathcal{L}$.*

Let G be a 2-connected plane graph and let $S \subseteq V(R_G)$. We say that a linkage $\mathcal{L}$ in G is an *S-linkage*, if $T(\mathcal{L}) \subseteq S$ and for every $\{s,t\}$ in the pattern of $\mathcal{L}$, s,t are in the same connected component of $G[S]$. Given a $z \in \mathbb{N}$ and a strongly connected normal subset S of $V(R_G)$, we define

$$B_G^{(\leq z)}(S) = V(G) \cap \bigcup\{X_x \mid \mathbf{dist}_Q(r_Q, x) \leq z\},$$

where $(\mathcal{X}, Q)$ is the LDAG-decomposition of R_G with respect to $\{S\}$.

Lemma 4.1.2. *Let G be a 2-connected plane graph and S be a strongly connected normal subset of $V(R_G)$. If G contains an S-linkage $\mathcal{L}$ of size at most k, then there is a linkage $\mathcal{L}'$ in $G[B_G^{(\leq z)}(S)]$ that is equivalent to $\mathcal{L}$, where $z = 2 \cdot f(k)$.*

Proof. Let $R = V(G) \setminus B_G^{(\leq z)}(S)$ and let $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\{S\}$. Let $f_1, \ldots, f_q$ be the even vertices of Q whose distance from r_q in Q is z. For each $i \in [q]$, let R_i be the set of v-vertices in the $\mathcal{X}$-suffix of f_i. Notice that $\bigcup_{i\in[q]} R_i = R$.

Let $G^{(i)} = G \setminus \bigcup_{j\in[i]} R_i$. Let also $(\mathcal{X}^{(i)}, Q^{(i)})$ be the LDAG-decomposition of $R_{G^{(i)}}$ with respect to $\{S\}$. We also denote $G^{(0)} := G$, $\mathcal{L}_0 := \mathcal{L}$, and $(\mathcal{X}^{(0)}, Q^{(0)}) := (\mathcal{X}, Q)$. Notice that $Q^{(i)}$ is obtained by $Q^{(i-1)}$ after replacing the $Q^{(i-1)}$-suffix of f_i by a single vertex f_i'. Observe that, in $(\mathcal{X}^{(i)}, Q^{(i)})$, the set $X_{f_i'}$ is a singleton containing the f-vertex of $R_{G^{(i)}}$ corresponding to the face of $G^{(i)}$ that is equal to the union of all faces of $G^{(i-1)}$ that are incident to a vertex in R_i. Notice that since the $Q^{(i-1)}$-suffix of f_i is strongly connected (because of Lemma 3.2.5), this union is indeed a face of $G^{(i)}$. Moreover, for the same reason, the boundary of this face of $G^{(i)}$ is a cycle, therefore $G^{(i)}$ remains 2-connected for every $i \in [q]$.

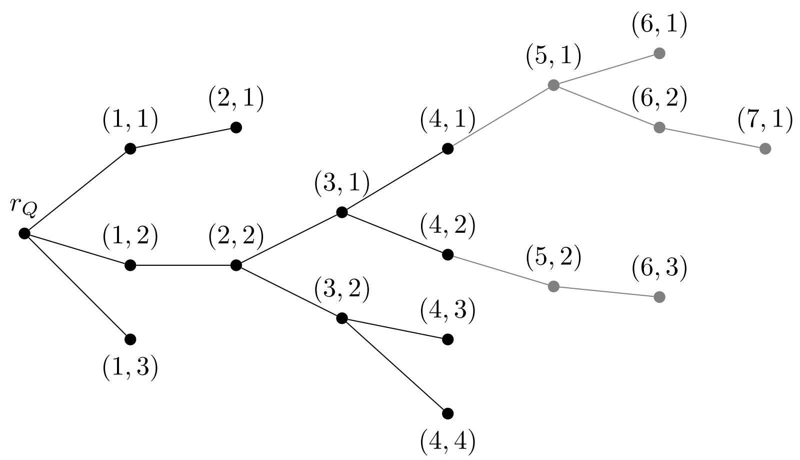

Figure 4.1: The graph Q as in Figure 3.5, where $f_1 = (4,1)$, $f_2 = (4,2)$, $f_3 = (4,3)$, and $f_4 = (4,4)$, while R_1 is the set of v-vertices in the $\mathcal{X}$-suffix of f_1 (i.e., vertices in $X_{(5,1)}$ and $X_{(7,1)}$), R_2 is the set of v-vertices in the $\mathcal{X}$-suffix of f_2 (i.e., vertices in $X_{(5,2)}$), and $R_3 = R_4 = \emptyset$.

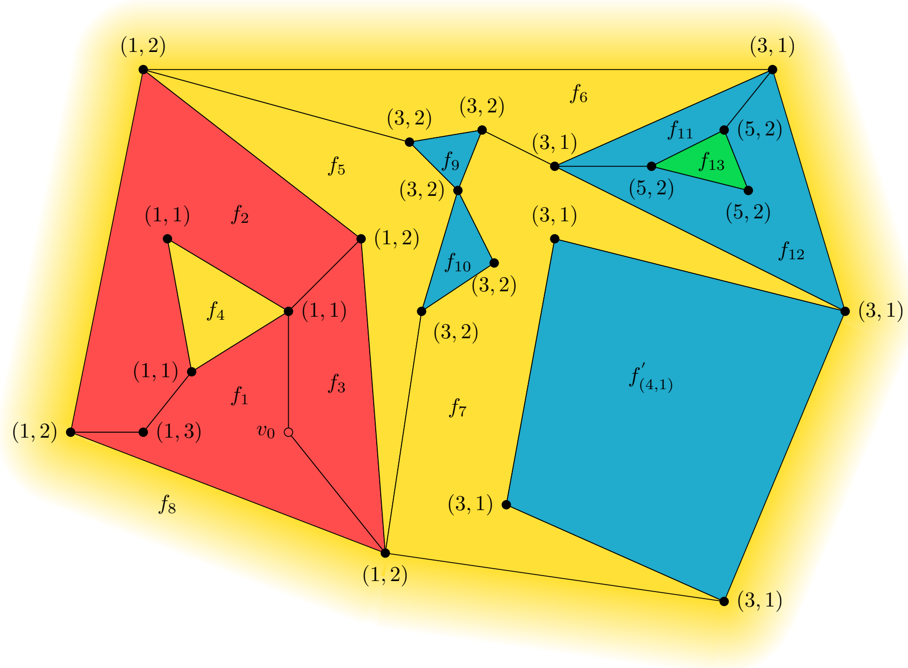

Figure 4.2: The graph $G^{(1)} = G \setminus R_1$, where G is the graph in Figure 3.4, and R_1 is the set of v-vertices in the $\mathcal{X}$-suffix of $(4,1)$, (as in the example in Figure 4.1). Again, the indices in the vertices correspond to the sets $X^{(1)}_{i,j}$ of $\mathcal{X}^{(1)}$ while same-colored faces are in the same (even) layer of $(\mathcal{X}^{(1)}, Q^{(1)})$.

Let $i \in [q]$ and let $\mathcal{L}_{i-1}$ be an S-linkage in $G^{(i-1)}$. We claim that

$$\textbf{ldist}_{G^{(i-1)}}(T(\mathcal{L}_{i-1}), R_i) \geq f(k).$$

Consider the path P in $Q^{(i-1)}$ joining f_i and a neighbor of $r_{Q^{(i-1)}}$ and observe that P has length $2\cdot f(k)-1$. Then, by Lemma 3.4.1, $G^{(i-1)}$ contains a sequence $C_1, \ldots, C_{f(k)}$ of nested cycles such that for every $j \in [f(k)]$ S is inside C_j and R_i is outside $C_{f(k)}$. Therefore, $T(\mathcal{L}_{i-1})$, as a subset of S, is inside C_1 and thus, $\textbf{ldist}_{G^{(i-1)}}(T(\mathcal{L}_{i-1}), R_i) \geq f(k)$. The claim follows.

By applying Proposition 4.1.1 to the graph $G^{(i-1)}$ the S-linkage $\mathcal{L}_{i-1}$, and the set R_i, we deduce the existence of an S-linkage $\mathcal{L}_i$ in $G^{(i)}$ that is equivalent to $\mathcal{L}_{i-1}$.

The lemma follows as $G^{(q)} = G[B^{(\leq z)}_G(S)]$, by setting $\mathcal{L}' = \mathcal{L}_q$. ☐

The next lemma is the main combinatorial result of this thesis and establishes the existence of an irrelevant set.

Lemma 4.1.3. *Let G be a plane graph, let $\mathcal{R} = \{R_1, \ldots, R_m\}$ be a root collection of R_G and let $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\mathcal{R}$. Let also $D_0, \ldots, D_\ell$ be the layers of $(\mathcal{X}, Q)$. If G contains a* $\bigcup\mathcal{R}$*-linkage $\mathcal{L}$ of size at most k then $G[V(G) \cap D_{\leq z}]$ contains a linkage $\mathcal{L}'$ that is equivalent to $\mathcal{L}$, where $z = 2 \cdot f(k) \cdot m$.*

Proof. Let $\mathcal{L}$ be a $\bigcup\mathcal{R}$-linkage in G of size at most k. Assume that $\ell > 2\cdot f(k)\cdot m$. Then, by Observation 3.1.1, there exists a non-negative even integer $p \leq 2 \cdot f(k) \cdot (m-1)$

such that none of the levels $L_{p+1}, \dots, L_{p+2\cdot f(k)}$ of Q contains a fusion vertex. It is enough to prove that $G[V(G) \cap D_{\leq p+2\cdot f(k)}]$ contains a linkage $\mathcal{L}'$ that is equivalent to L.

Let $S_1, \dots, S_q$ be the vertex sets of the connected components of $R_G[L_{\leq p}]$. Observe that, since p is even, then $L_{p+1} \subseteq V(G)$. Therefore, every $S_i, i \in [q]$ is a connected normal subset of $V(R_G)$ and by Lemma 3.2.5, it is also strongly-connected.

Let $\mathcal{T}$ be the pattern of $\mathcal{L}$ and let $\mathcal{T}_i = \mathcal{P} \cap (V(A_i))^2, i \in [q]$. Notice that $\{\mathcal{T}_1, \dots, \mathcal{T}_q\}$ is a partition of $\mathcal{T}$.

Also, for every $i \in [q]$, we consider the subgraph $\mathcal{L}_i$ of $\mathcal{L}$ whose pattern is $\mathcal{T}_i$ and observe that $\mathcal{L}_i$ is an S_i-linkage of G of size at most k. Therefore, for every $i \in [q]$, by Lemma 4.1.2, there is an S_i-linkage $\mathcal{L}_i'$ in $G[B_G^{(\leq 2\cdot f(k))}(S_i)]$ that is equivalent to $\mathcal{L}_i$. Notice that, due to the absence of fusion vertices in Q, the sets $V(\mathcal{L}_1'), \dots, V(\mathcal{L}_q')$ are pairwise disjoint.

Since $\{\mathcal{T}_1, \dots, \mathcal{T}_q\}$ is a partition of $\mathcal{T}$, then $\mathcal{L}' := \bigcup_{i\in[q]} \mathcal{L}_i'$ and $\mathcal{L}$ have the same pattern. Furthermore, we have that $\mathcal{L}'$ is a linkage in $G[\bigcup_{i\in[q]} B_G^{(\leq 2\cdot f(k))}(S_i)]$ and since $\bigcup_{i\in[q]} B_G^{(\leq 2\cdot f(k))}(S_i) \subseteq V(G) \cap D_{\leq p+2\cdot f(k)}$, then $\mathcal{L}'$ is a linkage in $G[V(G) \cap D_{\leq p+2\cdot f(k)}]$. Thus, the proof of the Lemma is complete. □

A *shortest path* in a graph G is a subgraph of G that is a path P and with the property that every path in G that has the same endpoints as P has no less edges than the edges of P.

Let G be a 2-connected plane graph, let $\mathcal{Z} = \{P_1, \dots, P_k\}$ be a collection of shortest paths of R_G. Notice that $V_i = N_{R_G}[V(P_i)]$ is a connected normal subset of $V(R_G)$. We now consider the graph $R_G[\bigcup_{i\in[k]} V_i]$ and observe that the vertex sets $\mathcal{R} = \{R_1, \dots, R_m\}$ of the connected components of $R_G[\bigcup_{i\in[k]} V_i]$ are also connected normal subsets of $V(R_G)$. Notice also that $\mathcal{R}$ is a root collection of R_G. We call $\mathcal{R}$ the *root collection of R_G generated by $\mathcal{Z}$*.

The next proposition follows from [6, Theorem 6].

Proposition 4.1.4. *Let G be a 2-connected plane graph, let $\mathcal{Z}$ be a collection of shortest paths in R_G and let $\mathcal{R}$ be the root collection R_G generated by $\mathcal{Z}$. Let also $(\mathcal{X}, Q)$ be the LDAG-decomposition of R_G with respect to $\mathcal{R}$ and $D_0, \dots, D_\ell$ be the layers of $(\mathcal{X}, Q)$. For every $z \in \mathbb{N}$ it holds that* $\mathbf{tw}(G[V(G) \cap D_{\leq z}]) = O(z)$.

CHAPTER 5

A LINEAR ALGORITHM FOR PDPP

The Planar Disjoint Paths **problem.** The problem that we examine in this thesis is the following.

Planar Disjoint Paths (PDPP)
Input: A planar graph G, and a collection $\mathcal{T} = \{(s_i, t_i) \in V(G)^2, i \in \{1,\ldots,k\}\}$ of pairs of $2k$ terminals of G.
Question: Are there k pairwise vertex-disjoint paths $P_1,\ldots,P_k$ in G such that for $i \in \{1,\ldots,k\}$, P_i has endpoints s_i and t_i?

We call the k-pairwise vertex-disjoint paths certifying a YES-instance of PDPP a *solution* of PDPP for the input $(G,\mathcal{T})$. We say that two instances $(G,\mathcal{T})$ and $(G',\mathcal{T}')$ of PDPP are *equivalent* if $(G,\mathcal{T})$ is a YES-instance of PDPP iff $(G',\mathcal{T}')$ is a YES-instance of PDPP.

We now present the main algorithmic result of this thesis.

Theorem 5.0.1. *There exists an algorithm that, given an instance $(G,\mathcal{P})$ of PDPP, where G is an n-vertex graph and $|\mathcal{P}| = k$, either reports that $(G,\mathcal{P})$ is a NO-instance or outputs a solution of PDPP for $(G,\mathcal{P})$. This algorithm runs in $2^{2^{O(k)}} \cdot n$ steps.*

The proof of Theorem 5.0.1 is based on the following.

Theorem 5.0.2. *There exists an algorithm that, given an instance $(G,\mathcal{T})$ of PDPP it outputs, in $O(|V(G)|)$ steps, a subgraph H of G, such that $(G,\mathcal{T})$ and $(H,\mathcal{T})$ are equivalent instances of PDPP and* $\mathbf{tw}(H) = 2^{O(k)}$.

Proof. Let $\mathcal{P} = \{\{s_1,t_2\},\ldots,\{s_k,t_k\}\}$. We first prove the theorem in the case where G is 2-connected. Let $\mathcal{Z} = \{P_1,\ldots,P_k\}$ be a collection of shortest paths of R_G such that $\{s_i,t_i\}$ are the endpoints of the path P_i for $i \in [k]$. Let also $\mathcal{R}$ be the root collection R_G generated by $\mathcal{Z}$, and let $(\mathcal{X},Q)$ be the LDAG-decomposition of R_G with respect to $\mathcal{R}$. Clearly $|\mathcal{R}| \leq k$. We set $G' := G[V(G) \cap D_{\leq z}]$ where $z := 2 \cdot f(k) \cdot |\mathcal{R}|$. Notice that G' is a subgraph of G that, from Proposition 4.1.4 has treewidth $O(k \cdot f(k)) = 2^{O(k)}$.

Moreover, because of Lemma 4.1.3, $(G, \mathcal{T})$ and $(G', \mathcal{T})$ are equivalent instances of PDPP.

We now deal with the case where G is not 2-connected. If G contains a leaf block B such that every vertex in $V(B)$ different than its cut vertex c is not a terminal, then we observe that $(G, \mathcal{T})$ and $(G \setminus (V(B) \setminus \{c\}), \mathcal{T})$ are equivalent instances of PDPP. This permits us to assume that each leaf block of G contains some terminal that is different from its cut-vertex. This implies that G contains at most $2k$ leaf blocks. We next describe two transformations on a graph G.

Firstly, if G contains a block B without any terminal and with exactly two cut-vertices c_1 and c_2 then we remove from G the vertices in $V(B) \setminus \{c_1, c_2\}$ and add the edge $\{c_1, c_2\}$. Also, if G contains a non-terminal cut-vertex c with exactly two neighbors, then we remove c and make adjacent its neighbors.

Let G_1 be the graph obtained by G after applying the two transformations until this is not possible any more. Notice that G_1 is a topological minor of G and that $(G, \mathcal{T})$ and $(G_1, \mathcal{T})$ are equivalent instances. Moreover, it is easy to observe that G_1 contains $O(k)$ blocks.

We say that two blocks B_1, B_2 of a graph G are *neighboring* if there is a face in G whose boundary contains an edge $e_1 = \{x_1, y_1\} \in E(B_1)$ and an edge $e_2 = \{x_2, y_2\} \in E(B_2)$. The operation of *joining* two neighboring blocks consists of adding either the edges $\{x_1, x_2\}$ and $\{y_1, y_2\}$ or the edges $\{x_1, y_2\}$ and $\{y_1, x_2\}$ so that the resulting graph embedding remains plane (if one of these edges is a loop, then do not add it). The construction of the resulting graph is completed by subdividing once each of the new edges.

Let G_2 be the graph obtained by G_1 after applying joins of neighboring blocks as long as this is possible. We denote by D the set of subdivision vertices and, given the instance $(G_1, \mathcal{T})$ of PDPP, we construct the instance $(G_2, \mathcal{T}')$ where $\mathcal{T}' = \mathcal{T} \cup \{\{d, d\} \mid d \in D\}$. Notice that $(G_1, \mathcal{T})$ and $(G_2, \mathcal{T}')$ are equivalent instances. Also, observe that G_2 is 2-connected and that $|\mathcal{T}'| = O(k)$. We refer to the subdivision vertices that where added during this process as *dummy terminals*.

As the theorem holds for the 2-connected case, there is a subgraph G_3 of G_2 such that $(G_2, \mathcal{T}')$ and $(G_3, \mathcal{T}')$ are equivalent instances and moreover $\mathbf{tw}(G_3) = 2^{O(k)}$. If we now remove from G_3 the dummy terminals, we obtain a graph G_4 such that $(G_4, \mathcal{T})$ and $(G_3, \mathcal{T}')$ are again equivalent instances. Notice now that G contains a subgraph H that is a subdivision of G_4 and such that none of its subdivision vertices is a terminal in $\mathcal{T}$. This implies that $(H, \mathcal{T})$ and $(G_4, \mathcal{T})$ are again equivalent instances. Moreover, as H is a subdivision of G_4 it also holds that $\mathbf{tw}(H) = 2^{O(k)}$. Therefore, the algorithm computes H according to the above steps and outputs $(H, \mathcal{T})$ as an equivalent instance of PDPP. □

The proof of Theorem 5.0.1 follows directly from Theorem 5.0.2 and the following result by Petra Scheffler.

Proposition 5.0.3 ([29]). *There exists an algorithm that, given an instance $(G, \mathcal{T})$ of PDPP and a tree decomposition of G of width at most w, either reports that $(G, \mathcal{T})$ is a NO-instance or outputs a solution of PDPP for $(G, \mathcal{T})$ in $2^{O(w \log w)} \cdot n$ steps.*

CHAPTER 6

CONCLUSION

In this thesis we tried to shed some light, from an algorithmic scope, to the study of planar graphs. We proved some structural results concerning plane graphs and provided a linear parameterized algorithm for PDPP. Meanwhile, towards further improving the parametric dependence for PDPP, an important breakthrough was achieved by Lokshtanov, Misra, Pilipczuk, and Saurabh [21] who recently announced an algorithm that runs in $2^{O(k)}n^{O(1)}$ steps. This result bypasses the irrelevant vertex technique by combining techniques from [3], cohomology techniques by Schrijver in [30] and ideas used in [5] for solving the disjoint paths problem on planar directed graphs. All these come at the cost of a higher, non-linear, polynomial contribution in n. While these results are already far-reaching, their further improvement towards a linear algorithm would be important as it would achieve two-fold algorithmic optimality both in the contribution of k and n in the running time. Also, it would be interesting to extend existing algorithmic results in the context of embedded graphs. To conclude, we believe that the decompositions mentioned in this thesis together with the application of the irrelevant vertex technique can be useful in other variants of DPP.

BIBLIOGRAPHY

[1] Isolde Adler, Martin Grohe, and Stephan Kreutzer. Computing excluded minors. In *19th annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 641–650. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[2] Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Tight bounds for linkages in planar graphs. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011*, volume 6755 of *Lecture Notes in Computer Science*, pages 110–121. Springer, Berlin, 2011.

[3] Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Irrelevant vertices for the planar Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 122:815–843, 2017.

[4] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[5] Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. The planar directed k-vertex-disjoint paths problem is fixed-parameter tractable. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 197–206, 2013.

[6] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Contraction decomposition in h-minor-free graphs and algorithmic applications. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 441–450, New York, NY, USA, 2011. ACM.

[7] Fedor V. Fomin, Saket Saurabh Daniel Lokshtanov, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on H-minor-free graphs. In *23st ACM–SIAM Symposium on Discrete Algorithms (SODA 2012)*. ACM-SIAM, San Francisco, California, 2012.

[8] A. Frank. Packing paths, cuts and circuits - a survey. In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, editors, *Paths, Flows and VLSI-Layout*, pages 49–100. Springer-Verlag, Berlin, 1990.

[9] András Frank. Connectivity and network flows. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics (Vol. 1)*, pages 111–177. MIT Press, Cambridge, MA, USA, 1995.

[10] Stavros G. Kolliopoulos. Edge-disjoint paths and unsplittable flow. In *Handbook of Approximation Algorithms and Metaheuristics*. CRC press, 2018. 2nd edition.

[11] Petr A. Golovach, M. Kamiński, D. Paulusma, and D. M. Thilikos. Induced packing of odd cycles in a planar graph. In *20th International Symposium on Algorithms and Computation (ISAAC 2009)*, volume 5878 of *LNCS*, pages 514–523. Springer, Berlin, 2009.

[12] Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, (STOC 2011)*, pages 479–488, 2011.

[13] Ken ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424 – 435, 2012.

[14] Marcin Kamiński and Dimitrios M. Thilikos. Contraction checking in graphs on surfaces. In *29th International Symposium on Theoretical Aspects of Computer Science, (STACS)*, pages 182–193, 2012.

[15] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[16] Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 2011.

[17] Ken-ichi Kawarabayashi, Stephan Kreutzer, and Bojan Mohar. Linkless and flat embeddings in 3-space and the unknot problem. In *Proceedings of the 2010 annual symposium on Computational geometry*, SoCG '10, pages 97–106, New York, NY, USA, 2010. ACM.

[18] Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: the unique linkage theorem. In *42nd ACM Symposium on Theory of Computing, STOC 2010*, pages 687–694. ACM, 2010.

[19] Yusuke Kobayashi and Ken-ichi Kawarabayashi. Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In *20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009)*, pages 1146–1155. ACM-SIAM, 2009.

[20] Mark R. Kramer and Jan van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. *Advances in Comp. Research*, 2:129–146, 1984.

[21] Daniel Lokshtanov, Pranabendu Misra, Michal Pilipczuk, and Saket Saurabh. Toppling the tower: An exponential time parameterized algorithm for planar disjoint paths. Personal communication.

[22] James F. Lynch. The equivalence of theorem proving and the interconnection problem. *ACM SIGDA Newsletter*, 5:31–36, 1975.

[23] Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012.

[24] Matthias Middendorf and Frank Pfeiffer. On the complexity of the disjoint paths problem. *Combinatorica*, 13(1):97–107, 1993.

[25] Bruce A. Reed, Neil Robertson, Alexander Schrijver, and Paul D. Seymour. Finding disjoint trees in planar graphs in linear time. In *Graph Structure Theory, Proceedings of AMS-IMS-SIAM Joint Summer Research Conference on Graph Minors held June 22 to July 5, 1991, at the University of Washington, Seattle*, pages 295–301. AMS-IMS-SIAM, 1991.

[26] Neil Robertson and Paul D. Seymour. Graph Minors. XIII. The disjoint paths problem. *J. Comb. Theory Ser. B*, 63(1):65–110, 1995.

[27] Neil Robertson and Paul D. Seymour. Graph minors. XXI. Graphs with unique linkages. *J. Comb. Theory Ser. B*, 99(3):583–616, 2009.

[28] Neil Robertson and Paul D. Seymour. Graph minors. XXII. Irrelevant vertices in linkage problems. *J. Comb. Theory, Ser. B*, 102(2):530–563, 2012.

[29] Petra Scheffler. A practical linear time algorithm for disjoint paths in graphs with bounded tree-width. Technical Report 396/1994, FU Berlin, Fachbereich 3 Mathematik, 1994.

[30] Alexander Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM J. Comput.*, 23(4):780–788, 1994.

[31] Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency (3 volumes)*, volume 24. Springer, 2003.

[32] P. D. Seymour. Disjoint paths in graphs. *Discrete Math.*, 29(3):293–309, January 1980.

[33] Yossi Shiloach. A polynomial solution to the undirected two paths problem. *J. ACM*, 27(3):445–456, 1980.

[34] Carsten Thomassen. 2-linked graphs. *Eur. J. Comb.*, 1:371–378, 1980.

[35] Jens Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Appl. Math.*, 61(1):83–90, 1995.