Plane and Simple: Using Planar Subgraphs for Efficient Algorithms

A dissertation submitted towards the degree
Doctor of Natural Science
of the Faculty of Mathematics and Computer Science of
Saarland University

by Andreas Schmid

Saarbrücken / 2019

Day of Colloquium:	2. December 2019
Dean of the Faculty:	Prof. Dr. Sebastian Hack
Chair of the Committee:	Prof. Dr. Markus Bläser
Reporters	
First reviewer:	Prof. Dr. Dr. h.c. Kurt Mehlhorn
Second reviewer:	Prof. Dr. Parinya Chalermsook
Academic Assistant:	Dr. Antonios Antoniadis

Abstract

Abstract In this thesis, we showcase how planar subgraphs with special structural properties can be used to find efficient algorithms for two NP-hard problems in combinatorial optimization.

In the first part, we develop algorithms for the computation of Tutte paths and show how these special subgraphs can be used to efficiently compute long cycles and other relaxations of Hamiltonicity if we restrict the input to planar graphs. We give an $O(n^2)$ time algorithm for the computation of Tutte paths in circuit graphs and generalize it to the computation of Tutte paths between any two given vertices and a prescribed intermediate edge in 2-connected planar graphs.

In the second part, we study the Maximum Planar Subgraph Problem (Mps) and show how dense planar subgraphs can be used to develop new approximation algorithms for this problem. All new algorithms and arguments we present are based on a novel approach that focuses on maximizing the number of triangular faces in the computed subgraph. For this, we define a new optimization problem called Maximum Planar Triangles (Mpt). We show that this problem is NP-hard and quantify how good an approximation algorithm for Mpt performs as an approximation for Mps. We give a greedy $\frac{1}{11}$-approximation algorithm for Mpt and show that the approximation ratio can be improved to $\frac{1}{6}$ by using locally optimal triangular cactus subgraphs.

Zusammenfassung In dieser Dissertation zeigen wir, wie planare Teilgraphen mit speziellen Eigenschaften verwendet werden können, um effiziente Algorithmen für zwei NP-schwere Probleme in der kombinatorischen Optimierung zu finden.

Im ersten Teil entwickeln wir Algorithmen zur Berechnung von Tutte-Wegen und zeigen, wie diese verwendet werden können, um lange Kreise und andere Lockerungen der Hamilton-Charakteristik zu finden, wenn wir uns auf Graphen in der Ebene beschrnken. Wir beschreiben zunächst einen $O(n^2)$-Algorithmus in Circuit-Graphen und verallgemeinern diesen anschließend für die Berechnung von Tutte-Wegen in 2-zusammenhngenden planaren Graphen.

Im zweiten Teil untersuchen wir das Maximum Planar Subgraph Problem (Mps) und zeigen, wie besonders dichte planare Teilgraphen verwendet werden können, um neue Approximationsalgorithmen zu entwickeln. Unsere Ergebnisse basieren auf einem neuartigen Ansatz, bei dem die Anzahl der dreieckigen Gebiete im berechneten Teilgraphen maximiert wird. Dazu definieren wir ein neues Optimierungsproblem namens Maximum Planar Triangles (Mpt). Wir zeigen, dass dieses Problem NP-schwer ist und quantifizieren, wie gut ein Approximationsalgorithmus für Mpt als Approximation für Mps funktioniert. Wir geben einen $\frac{1}{11}$-Approximationsalgorithmus für Mpt und zeigen, wie dies durch die Verwendung von lokal optimaler Kaktus-Teilgraphen auf $\frac{1}{6}$ verbessert werden kann.

Acknowledgments

I want to thank Parinya Chalermsook for being a great adviser during my whole Ph.D. and for always valuing my personal happiness as high as my professional interests. I also need to thank Kurt Mehlhorn who was the first person in academia who saw my unconventional CV and thought of it as interesting and not as a drawback. These two advisers created an environment that allowed me to find research topics for which I have a real passion and develop a way of conducting research that always keeps me motivated. Another important role in my research career was played by Jens Schmidt, who introduced me not only to the world of Graph Theory but especially to Tutte paths which turned out to be one of the main topics in my research.

I am also very grateful to have had friends like Daniel, Gorav, Pavel, and Thatchaphol during my studies and the process of writing this dissertation. Without them, I would probably have given up on my algorithms and theory courses long before starting a Ph.D. in theoretical computer science. Even after we finished our courses I could always rely on you to bounce off new research ideas or discuss related topics.

Contents

CHAPTER 1

Introduction

Many combinatorial optimization problems that appear in the real world are commonly modeled using graphs. For example, finding the most profitable route for a traveling salesman or assigning medical students to teaching hospitals. These problems are among the oldest problems in algorithms and theoretical computer science in general. Once we model problems using graphs, we are often able to identify crucial structural properties of the underlying graphs that allow us to build provably efficient algorithms. Unfortunately, for many of the most famous combinatorial problems, it is not possible to find an efficient algorithm that computes the optimal solution for every input (unless P=NP). As efficiency often has the highest priority in practice, we either have to sacrifice the optimality of the output or build algorithms that can only handle a subset of all possible input instances. In this thesis, we explore two approaches for handling NP-hard problems and combine them with the study of structural graph theory to make two notoriously hard problems accessible in practice.

In the first approach, we restrict the input to a less general set of instances and then try to prove that the problem is less difficult on that set of inputs. Here we investigate the longest cycle problem together with some of its relaxations, and we restrict the input to planar graphs. The largest possible solution to this problem in a given graph is a cycle that goes through every vertex exactly once. Such a cycle is called a Hamiltonian cycle and we say a graph is Hamiltonian if it contains a Hamiltonian cycle. The longest cycle problem is a special case of the traveling salesman problem; if all cities that he may visit yield the same profit and all roads that the salesman takes to get there have the same cost, then the problem reduces to finding the longest route without having the salesman visit any place twice. This problem is known to be NP-hard and it would be interesting to know for which set of planar graphs we can prove the existence of a cycle of a certain length as we already know that not all planar graphs are Hamiltonian. In addition, we would want to know whether there exists an efficient algorithm to compute it.

The graph structure used in this thesis to attack the longest cycle problem in the described manner is called a Tutte Path. At its core, a Tutte Path gives us the information we need to split the original problem into smaller subproblems and then combine their solutions into a solution for the original input graph. Part one of this Thesis covers the algorithmic complexity of finding Tutte paths in planar graphs and highlights some algorithmic applications in the context of finding long cycles.

The second approach to handling NP-hard problems in practice is to design fast algorithms whose output is not necessarily optimal but guarantees a certain quality. Such algorithms are known as approximation algorithms. Our focus here lies on graph drawing problems. The most famous such problem is probably the crossing number problem, in which we are given a graph and want to draw it with as few edge-crossings as possible. As for many other graph drawing problems, the best-known algorithmic strategy relies on first finding a large planar subgraph of the input graph and then drawing the remaining graph. This problem alone is known as the Maximum Planar Subgraph Problem and will be in the focus of the second part of this Thesis. Here, the linear matroid parity problem plays an important role. As a generalization of the matching problem, algorithms for linear matroid parity can also be used to solve the previously mentioned assignment of medical students to teaching hospitals. We will inspect various subgraph structures and different ways to assemble them from the input graph and analyze their usability for approximating the Maximum Planar Subgraph problem.

PART I

Computing Tutte Paths in Polynomial Time

This part is the result of a close collaboration with Jens M. Schmidt. It is based on two articles. The first was published in the proceedings of the *Symposium on Theoretical Aspects in Computer Science (STACS) 2015* [61], and the journal *ACM Transactions on Algorithms* [62]. The second article was published in the proceedings of the *45th International Colloquium on Automata, Languages, and Programming (ICALP) 2018* [63].

CHAPTER 2

Introduction to Tutte Paths

The question of whether a graph $G = (V, E)$ is *Hamiltonian* is among the most fundamental graph problems. Whitney [74] proved that every 4-connected maximal planar graph is Hamiltonian. A connected graph is called k-connected for some positive integer k if we have to remove at least k vertices to disconnect it and maximal planar if adding any edge would make it non-planar. Tutte extended this to arbitrary 4-connected planar graphs by showing that every 2-connected planar graph contains a Tutte path [72, 73]. Figure 2.1 shows how the 2-connected, 4-connected, and the other classes of planar graphs mentioned in this chapter are related. Unfortunately, there are numerous examples proving that 3-connected planar graphs are not necessarily Hamiltonian; in fact, even deciding whether a 3-connected 3-regular planar graph is Hamiltonian is NP-hard [32]. In general, one may ask how "close" 3-connected planar graphs are to Hamiltonicity. As a result many relaxed notions of Hamiltonicity have been studied in the past.

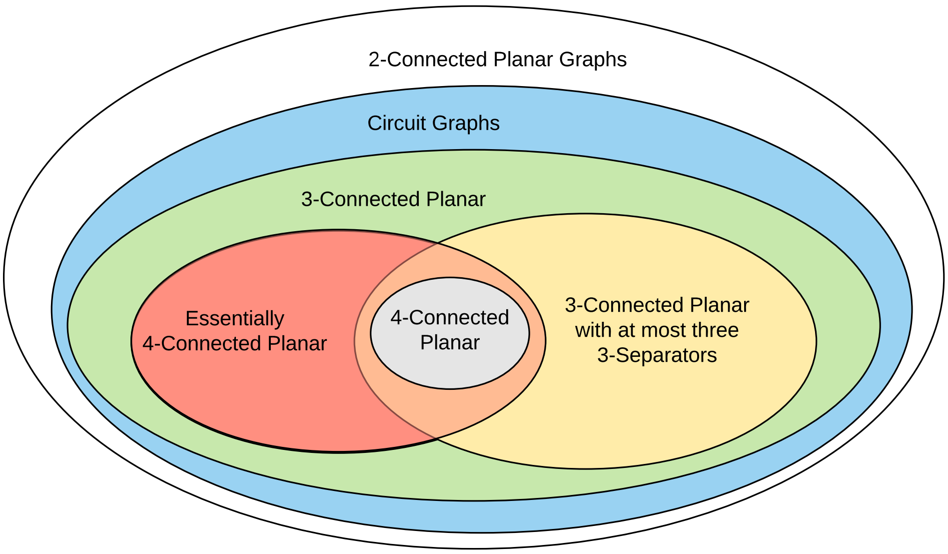


Figure 2.1: A diagram of the relation between classes of planar graphs that we consider in this thesis. The 2-connected planar graphs are the most general class, while the 4-connected planar graphs form the most restricted class.

A *k-walk* is a walk that visits every vertex in a graph at least once and at most k times (edges may be visited multiple times). A walk is called *closed* if it has the same start- and endvertex. Thus, a closed 1-walk is a Hamiltonian cycle. Jackson and Wormald conjectured in [38] that every 3-connected planar graph contains a closed 2-walk. In a seminal result [29], Gao and Richter proved this conjecture in 1994 in the affirmative. Barnette [2] proved that every 3-connected planar graph contains a *3-tree*, i.e., a spanning tree with maximum degree at most three, while a Hamiltonian path is equivalent to a 2-tree. Interestingly, a 3-tree can be directly obtained from a closed 2-walk in linear time, as shown in [38, Lemma 2. 2(ii)]. By itself, a 3-tree can be computed in linear time as shown in the Ph.D.-thesis of Strothmann [65]. Biedl showed that 3-trees (and in fact, more special variants of them) can also be computed by canonical orderings [4]. Finally, one might try to prove that even if a given graph is not necessarily Hamiltonian, we can always find a cycle of at least a certain length. Jackson and Wormald [39] showed that every essentially 4-connected planar graph (we give the definition of these graphs in Section 2.4) contains a cycle of length at least $\frac{2n+4}{5}$. They also gave an upper bound by showing that there exists an infinite family of essentially 4-connected planar graphs, whose longest cycle has length $\frac{2n+8}{3}$.

For planar graphs and graphs embeddable on higher surfaces, *Tutte paths* have proven to be one of the most successful tools for attacking Hamiltonicity problems and problems on long cycles. For this reason, there is a wealth of existential results in which Tutte paths serve as the main ingredient; in chronological order, these are [73, 70, 67, 17, 58, 59, 68, 76, 40, 69, 31, 35, 47, 56, 55, 42, 61, 24, 7]. A central concept for Tutte paths is the notion of H-bridges (see [73] for some of their properties): For a subgraph H of a plane graph G with outer-face boundary C_G. an *H-bridge* of G is either an edge that has both endvertices in H but is not itself in H or a component K of $G - H$ together with all edges (and the endvertices of these edges) that join vertices of K with vertices of H. A vertex of an H-bridge L is an *attachment* of L if it is in H, and an *internal* vertex of L otherwise. An *outer* H-bridge of G is an H-bridge that contains an edge of C_G. A *Tutte path* (*Tutte cycle*) of a plane graph G, is a path (a *cycle*) P of G such that every outer P-bridge of G has at most two attachments and every P-bridge at most three attachments. Thomassen [70] proved the following generalization of Tutte's result, which also implies that every 4-connected planar graph with n vertices is Hamiltonian-connected, i.e. contains a path of length $n - 1$ between any two vertices.

Theorem 2.1 (Thomassen [70])**.** *Let G be a 2-connected plane graph, $x \in V(C_G)$, $\alpha \in E(C_G)$ and $y \in V(G) - x$. Then G contains a Tutte path from x to y through α.*

Sanders [59] then generalized Thomassen's result further by allowing to choose both endvertices of the Tutte path arbitrarily.

Theorem 2.2 (Sanders [59])**.** *Let G be a 2-connected plane graph, $x \in V(G)$, $\alpha \in E(C_G)$ and $y \in V(G) - x$. Then G contains a Tutte path from x to y through α.*

On top of the previously mentioned series of fundamental results, Tutte paths have been used in two research branches: while the first deals with the existence of Tutte paths on graphs embeddable on higher surfaces [67, 8, 68, 76, 69, 42], the second [38, 29, 8, 30, 40, 31, 53] investigates generalizations or specializations of Hamiltonicity such as *long cycles*, *Hamiltonian connectedness* and *k-walks.*

In [30, 31] Gao, Richter, and Yu published a refined decomposition that utilizes Tutte paths to give the existence of a special closed 2-walk, namely one in which every vertex visited twice is contained in a 3-separator. To achieve this, the authors proved the existence of a Tutte path T with T-bridges $B_1, B_2, \ldots, B_k$, for which a set $S = \{s_1, s_2, \ldots, s_k\}$ of vertices exists such that s_i is an attachment of B_i for each i. The set S is called *system of distinct representatives (SDR)* of the T-bridges. In fact, the result is shown for the class of circuit graphs, which contain all 3-connected planar graphs (illustrated in Figure 2.1); a *circuit graph* (G, C_G) is a plane graph G with a (simple) cycle C_G as outer-face boundary such that the following property is satisfied: For every vertex v in $G \setminus C_G$, G contains three independent paths from v to distinct vertices in C. We refer to this property by the *3-path property*.

Theorem 2.3 ([30, 31])**.** *Let (G, C_G) be a circuit graph, let $x, u, y \in V(C_G)$ with $x \neq y$ and let $a \in \{x, u\}$. Then there is a Tutte path P of G from x to y through u and an SDR S of the non-trivial P-bridges such that $a \notin S$.*

Theorem 2.3, as stated here, is slightly weaker than the one in [30, 31] (in which $y \in V(G)$), but is still sufficient to first find a Tutte cycle and then compute a closed 2-walk in any given circuit graph.

Corollary 2.4 ([30, 31])**.** *Let (G, C_G) be a circuit graph and let $x, y \in V(C_G)$. Then there is a Tutte cycle T of G and an SDR S of the non-trivial T-bridges in G with $x, y \in V(T)$ and $x, y \notin S$.*

The closed 2-walk constructed from the Tutte cycle given by Corollary 2.4 forms a Hamiltonian cycle if the graph is 4-connected and, hence, generalizes Tutte's theorem to 3-connected planar graphs.

Theorem 2.5 ([30, 31])**.** *Let (G, C_G) be a circuit graph and let $x, y \in V(C_G)$. Then there is a closed 2-walk W in G visiting x and y exactly once such that every vertex visited twice is contained in either a 2-separator or an internal 3-separator of G.*

Unfortunately, in all the results that utilize Tutte paths mentioned so far, very little is known about the complexity of actually finding a Tutte path. This is crucial, as the task of finding Tutte paths is almost always the only reason that hinders the computational tractability of the problem. The main obstruction so far is that Tutte paths are found by decomposing the given graph into overlapping subgraphs, on which induction is applied. Although this is enough to prove existence results, these overlapping subgraphs do not allow to bound the running time polynomially (as argued in [33]).

On the other hand, inspired by Tutte's classic result, Gouyou-Beauchamps [33] showed that a Hamiltonian cycle in a 4-connected planar graph can be computed in polynomial time. Asano, Kikuchi, and Saito showed that a Hamiltonian cycle can be computed in linear time when the 4-connected planar input graph is additionally maximal planar [1]. Thomassen claimed that one could also derive a polynomial time algorithm from his more general existence proof in [70]. In [17] it was shown that this statement was too optimistic, as the subgraphs arising from his decomposition may again overlap in big parts. Chiba and Nishizeki [18] showed that this problem can be avoided for 4-connected planar graphs and gave a linear time algorithm to compute a Hamiltonian cycle for these graphs.

The much more general problem of overlapping subgraphs when computing Tutte paths in 3-connected planar graphs has recently been resolved in [60] where it was shown how to extend the decomposition in [30, 31] to avoid big overlapping subgraphs. Only this year it was shown how to compute a Tutte path with both endvertices on the outer face of a given 2-connected plane graph in linear time Unfortunately, the authors point out that this approach cannot be used to give an algorithm for Sanders's result, which is mandatory for some of the previous mentioned existential results.

2.1 Our Results

Our motivation is two-fold. First, we want to make Tutte paths accessible to algorithms. We will show that Tutte paths can be computed in time $O(n^2)$ in any planar graph. This has an impact on almost all the applications using Tutte paths listed above. Second, we aim for computing the strongest possible known variant of Tutte paths, encompassing the many incremental improvements on Tutte paths made over the years. We will, therefore, develop an algorithm for Sander's existence result [59], which was proven to be the best possible in many aspects. For example, Sanders [59] showed that it is only possible to prescribe an edge if it is contained in C_G. Jackson et al. [40] showed that every circuit graph contains even a Tutte *cycle* through any two prescribed vertices and an edge on the outer face. However, Sander's result is still best-possible, as this cannot be expected from 2-connected graphs, as Figure 2.2 shows.

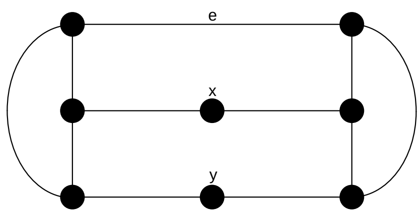


Figure 2.2: A 2-connected planar graph that has no Tutte cycle through x, y and e.

We show how to overcome the problem of overlapping subgraphs by extending the known decomposition for finding Tutte paths in planar graphs. We start with the decomposition given by Gao, Richter, and Yu for circuit graphs and modify it such that all arising subgraphs will be edge-disjoint. The new decomposition immediately yields a description of a polynomial time algorithm for computing Tutte paths in circuit graphs and allows us to bound its running time. This is captured in the following theorem.

Theorem 2.6. *Let (G, C_G) be a circuit graph, then the Tutte cycle T of Corollary 2.4 can be computed in time $O(n^2)$.*

This leads to a cubic time algorithm that computes the special closed 2-walk of [30, 31].

Theorem 2.7. *Let (G, C_G) be a circuit graph and let x, y be vertices of C_G. A closed 2-walk of G such that x and y are visited exactly once and every vertex visited twice is contained in either a 2-separator or an internal 3-separator of G can be computed in time $O(n^3)$.*

The relation between Tutte paths and 2-walks is highlighted in Section 2.4 as one of the examples on how to compute long cycles and relaxations using Tutte paths. The decomposition used for this was also part of [60], but we refine and simplify the decomposition resulting in an exact description of the algorithmic tasks necessary to be resolved. This, in turn, allows us to give an upper bound on the running time of the resulting algorithm. In [60] it was only shown that there must exist an algorithm to compute the special closed 2-walk as introduced in [30, 31].

We then move on to all 2-connected planar graphs by giving decompositions that refine the original ones used for Theorem 2.1 and Theorem 2.2, and allows to decompose a given graph into graphs that pairwise intersect in at most one edge. We show that this small overlap does not prevent us from achieving a polynomial running time for the computation of Tutte paths. All arising graphs in this decomposition will again be plane and simple. We proceed by showing how this decomposition can be computed efficiently in order to find the Tutte path of Theorem 2.2. Our main result in this part of the thesis is hence the following, giving the first polynomial time algorithm for computing Tutte paths as stated in Theorem 2.2 in any 2-connected planar graph.

Theorem 2.8. *Let G be a 2-connected plane graph, $x \in V(G)$, $\alpha \in E(C_G)$ and $y \in V(G) - x$. Then a Tutte path of G from x to y through α can be computed in time $O(n^2)$.*

Sanders's result has also an immediate extension to all connected planar graphs that contain a simple path from x to y through α [55], which can be computed simply and efficiently from our result by using block-cut trees. Chapter 4 presents the decomposition with small overlap that proves the existence of Tutte paths. On the way to Theorem 2.8, we give full algorithmic counterparts of the approaches of Thomassen and Sanders; for example, we describe small overlap variants of Theorem 2.1 and of the *Three Edge Lemma* [67, 58], which was used in the purely existential result of Sanders [59] as a black box.

2.2 Preliminaries

We assume familiarity with standard graph theoretic notations as in [23]. Let $deg(v)$ be the degree of a vertex v. We denote the subtraction of a graph H from a graph G by $G - H$ and the subtraction of a vertex or edge x from G by $G - x$.

A *k-separator* of a graph $G = (V, E)$ is a subset $S \subseteq V$ of size k such that $G - S$ is disconnected. A graph G is *k-connected* if $|V| > k$ and G contains no $(k-1)$-separator. For a path P and two vertices $x, y \in P$, let xPy be the smallest subpath of P that contains x and y. For a path P from x to y, let $inner(P) := V(P) - \{x, y\}$ be the set of its inner vertices. Paths that intersect pairwise at most at their endvertices are called *independent*.

A connected graph without a 1-separator is called a *block*. A *block of a graph G* is an inclusion-wise maximal subgraph of G that is a block. Every block of a graph is thus either 2-connected or has at most two vertices. It is well-known that the blocks of a graph partition its edge-set. A graph G is called a *chain of blocks* if it consists of blocks $B_1, B_2, \dots, B_k$ such that $V(B_i) \cap V(B_{i+1})$, $1 \leq i < k$, are pairwise distinct 1-separators of G and G contains no other 1-separator. In other words, a chain of blocks is a graph, whose block-cut tree [36] is a path.

A *plane* graph is a planar embedding of a graph. Let C be a cycle of a plane graph G. For two vertices x, y of C, let xCy be the clockwise path from x to y in C. For a vertex x and an edge e of C, let xCe be the clockwise path in C from x to the endvertex of e such that $e \notin xCe$ (define eCx analogously). Let the subgraph of G *inside* C be the subgraph induced by $E(C)$ and all edges intersecting the open disc-homeomorph of the plane interior of C.

Given a plane graph G, let C_G denote the boundary of its outer face. For vertices x, y and an edge $\alpha \in C_G$, let an *x-α-y-path* be a Tutte path from x to y that contains α. We may use *x-y-path*, for simplicity, to denote an *x-α-y-path* for which an arbitrarily edge $\alpha \in C_G$ can be chosen. We end this section with a simple observation on Tutte paths.

Observation 2.9. *Let T be a Tutte path of a 2-connected planar graph. If $|V(T)| \geq 4$, then the attachments of any T-bridge form a separator in G.*

2.3 Important Properties of Circuit Graphs

A subgraph inside a cycle of a 3-connected plane graph G is not necessarily 3-connected; however, its only 2-separators must have both vertices on the outer face. Since we will often use induction on such subgraphs when describing the decomposition, we will deal with circuit graphs instead of 3-connected plane graphs.

Equivalently to the 3-Paths property, a planar graph is a circuit graph if it can be obtained from a 3-connected graph by deleting a vertex. Clearly, circuit graphs are 2-connected and generalize 3-connected plane graphs. In the following, we will give several lemmas about circuit graphs that will be used throughout the first part of this thesis. The next two lemmas are probably folklore.

Lemma 2.10 ([60]). *Let $\{u, v\}$ be a 2-separator of a circuit graph (G, C_G). Every component of $G \setminus \{u, v\}$ contains a vertex of C_G.*

Proof. Assume to the contrary that $G \setminus \{u, v\}$ has a component K with $V(K) \cap V(C_G) = \emptyset$. Since K does not contain a vertex of C_G, each path from a vertex $w \in V(K)$ to C_G contains u or v. Thus, there are no three independent paths from w to C, contradicting the 3-Paths Property. □

Lemma 2.11 ([60]). *Let $\{u, v\}$ be a 2-separator of a circuit graph (G, C). Then u and v are contained in C and $G \setminus \{u, v\}$ has exactly two components.*

Proof. First, assume that u or v, say u, is not contained in C. As $\{u, v\}$ is a 2-separator of G, $G \setminus \{u, v\}$ has at least two components. Since $u \notin V(C)$, one component of $G \setminus \{u, v\}$ must contain all remaining vertices of C. This contradicts Lemma 2.10. For the second claim, observe that $G \setminus \{u, v\}$ has at most two components that contain vertices of C, as $C \setminus \{u, v\}$ is the union of at most two paths. Thus, a third component would contradict Lemma 2.10. □

Next, we state several lemmas on how a circuit graph can be decomposed into smaller circuit graphs.

Lemma 2.12 ([29]). *Let $\{u, v\}$ be a 2-separator of a circuit graph (G, C_G). For each nontrivial $\{u, v\}$-bridge H of G, $H \cup uv$ is a circuit graph.*

Lemma 2.13 ([29]). *Let C be any cycle in a circuit graph (G, C_G) and let H be the subgraph inside C. Then (H, C) is a circuit graph.*

A key idea in the decomposition of circuit graphs is that deleting a vertex of the outer face boundary results in a plane chain of blocks. Every block in this chain will either be just an edge or a circuit graph due to Lemma 2.13.

Lemma 2.14 ([29]). *Let (G, C) be a circuit graph and let $v \in V(C)$. Then $G \setminus v$ is a plane chain of blocks $B_1, B_2, \ldots, B_k$ and, if $k > 1$, one of the neighbors of v in C is in $B_1 \setminus B_2$ and the other is in $B_k \setminus B_{k-1}$.*

If the outer face boundary of the circuit graph is a triangle we can find an even more special structure.

Lemma 2.15 ([30, 60]). *Let (G, C) be a circuit graph such that $C = \{v, w, z\}$ is a triangle and $G \neq C$. Then $G \setminus v$ is a circuit graph and $G \setminus \{v, w\}$ is a plane chain of blocks $B_1, B_2, \ldots, B_k$ and, if $k > 1$, z is in $B_1 \setminus B_2$ and one neighbor of w is in $B_k \setminus B_{k-1}$.*

Proof. Due to Lemma 2.14, $G \setminus v$ is a plane chain of blocks with $z \in B_1$ and $w \in B_k$. According to the 3-Paths Property, G contains independent paths from every vertex in $G \setminus V(C)$ to v, w and z. Thus, $G' := G \setminus v$ is a block and therefore forms a circuit graph (G', C'). Applying Lemma 2.14 to (G', C') gives that $G' \setminus w$ is a plane chain of blocks with $z \in B_1$ and a neighbor of w in B_k. □

According to Lemma 2.10, both vertices of a 2-separator of any circuit graph must lie on the outer face boundary. The following lemma utilizes Observation 2.9 to strengthen this statement for the 2-separators that are attachments of T-bridges, for some Tutte path T of (G, C_G).

Lemma 2.16 ([60]). *Let (G, C) be a circuit graph with a Tutte path T from $x \in V(C)$ to $y \in V(C)$. Then every T-Bridge with two attachments has either both attachments on xCy or both on yCx.*

Proof. Assume otherwise. Let J be a T-bridge with two attachments $\{c, d\}$, $c \in xCy \setminus \{x, y\}$ and $d \in yCx \setminus \{x, y\}$. By Observation 2.9, $\{c, d\}$ is a 2-separator in G. Thus, $G \setminus \{c, d\}$ contains exactly two components X and Y with $x \in X$ and $y \in Y$ that cover $C \setminus \{c, d\}$, according to Lemma 2.11. Due to Lemma 2.10, X and Y must contain at least one vertex of C each. It follows that the inner vertex set of J is either X or Y. In both cases, J contains an edge of T, which contradicts that J is a T-bridge. □

2.4 Finding Long Cycles Using Tutte Paths

Several of the results mentioned in the introduction (for example [67, 58, 68, 42]) are constructive up to the point where they apply Theorem 2.1 or Theorem 2.2 on subgraphs when decomposing the given graph. Thus using our Algorithm from Theorem 2.8 as a subroutine immediately implies polynomial time algorithms where no efficient algorithms were published before. We present three other applications that illustrate how our result can be used on a 3-connected planar graph when we have various restrictions on the structure of its separators.

Long Cycles in Essentially 4-Connected Planar Graphs: A 3-separator S of a graph G is called trivial if $G \setminus S$ has at least one component that consists of exactly one vertex. A graph is called essentially 4-connected if it is 3-connected and each of its 3-separators is trivial. As mentioned in the introduction, Jackson and Wormald [39] showed that every essentially 4-connected planar graph contains a cycle of length at least

$\frac{2n+4}{5}$. In [24] Tutte paths were used to show that every essentially 4-connected graph contains a cycle of length at least $\frac{n+4}{2}$. This lower bound was further improved to $\frac{3n+6}{5}$ in [26], and the authors illustrate in a separate section how to use the algorithm from Theorem 2.8 to compute such a cycle in time $O(n^2)$. In a recently published preprint, the same authors [25] improved this lower bound even further to $\frac{5n+10}{8}$ and stated that the algorithmic description as given in [26] can be applied for the new result as well.

Hamiltonian Cycles in Graphs with at most two 3-Separators: In [7] it was shown that every 3-connected planar graph having at most three 3-separators is Hamiltonian. To achieve this result the authors use the result by Jackson and Yu [40], which in circuit graphs is stronger than Theorem 2.2. Unfortunately, at this point, we do not know of any polynomial time algorithm that computes the Tutte cycle as shown to exist by Jackson and Yu. Here we will show that if a 3-connected planar graph contains at most two 3-separators, then the algorithm from Theorem 2.8 can be used to compute a Hamiltonian cycle in time $O(n^2)$. The key idea is to ensure that the Tutte cycle we compute crosses each 3-separator of the given graph. In turn, we will show that there cannot exist any bridges of the computed Tutte cycle and thus that it actually is a Hamiltonian cycle.

Let G be a 3-connected planar graph with at most two 3-separators. If G does not have any 3-separator, then G is 4-connected, and we can use Theorem 2.8 (or the linear time algorithm from [18]) to compute a Hamiltonian cycle. This is based on the fact that the attachments of any bridge of the computed Tutte path would form a separator of order less than four, the existence of a bridge would therefore contradict the 4-connectivity of the given graph. Therefore, we assume that there exists at least one 3-separator and denote it by $A = \{u, v, w\}$. Any 3-connected planar graph has a unique embedding, thus when embedding G the only choice we have is which face of G serves as the outer face C_G. It is important to choose the outer-face carefully as Theorem 2.8 allows us to prescribe one edge of the outer-face to be contained in the computed Tutte cycle. How to choose this edge depends on whether there exists a second 3-separator in G. If A is the only 3-separator in G, then let a denote any vertex in $V \setminus \{u, v, w\}$ and a' an arbitrary neighbor of a in $G \setminus \{u, v, w\}$. If otherwise there exists a second 3-separator $B \neq A$ with vertices $\{x, y, z\}$ in G, then A and B can intersect in at most two vertices. We may assume that x is not in A and u is not in B. We will have to choose a and a' more carefully, in this case, to ensure that the Tutte Cycle we compute actually crosses both 3-separators of G. As G is 3-connected there are exactly two nontrivial A-bridges in G, otherwise, the three A-bridges and their common attachments would imply the existence of a $K_{3,3}$ minor in G, contradicting its planarity. Let e be any edge incident to u in the A-bridge of G that does not contain x. We choose any of the two faces incident to e as our outer-face C_G and embed G accordingly on the plane. Again, G has exactly two nontrivial B-bridges one of which contains u. Let a' denote any neighbor of a in the B-bridge of G that does not contain u. At least one such neighbor must exist.

By Theorem 2.8 we can find a Tutte path P from a to a' through e in $O(n^2)$ time. It remains to show that there does not exist any P-bridge in G, and therefore, $P + aa'$ forms a Hamiltonian cycle in G.

Theorem 2.17. *P is a Hamiltonian path in G.*

Proof. As G is 3-connected, any P-bridge of G must have three attachments. By Observation 2.9 any set of attachments is equal to a 3-separator in G. As G has at most two 3-separators A and B, it suffices to show that there does not exist a P-bridge of G with attachments equal to the vertices in A or B.

Assume for contradiction that there exists a P-bridge L of G with attachments $\{u, v, w\}$. As argued above, there are exactly two nontrivial A-bridges of G. By construction one of them J_e contains e and the other J_a contains a. We first show that L can not contain internal vertices of both J_a and J_e at the same time and thus must be a subset of either J_a or J_e. Assume otherwise that there are vertices $p, q \in L$, such that $p \in J_a$ and $q \in J_e$. By definition $L \setminus \{u, v, w\}$ must be a connected component, and therefore, there must be a path in $L \setminus \{u, v, w\}$ from p to q, which contradicts that A is a 3-separator of G. Without loss of generality, we assume that $L \subseteq J_a$, then note that as a was one of the prescribed vertices when computing P, we have that a is in P, and therefore, not in L. As G is 3-connected, there must be three independent paths from a to the endvertex of e not in A. Each one of these paths goes through a different vertex in A. In addition, these three paths can intersect L only in its attachments as otherwise there would exist a fourth attachment of L. Now we can construct a $K_{3,3,}$ from G by contracting all edges in these three independent paths except for the ones incident to the endvertices and the vertices in A and contracting J_e and L to one vertex each. This contradicts that G is a planar graph. If G contains a second 3-separator B, we can use the same argument as above for B, where a' would serve as a and u as the endvertex of e not in A. □

Corollary 2.18. *We can find a Hamiltonian cycle in graphs with at most two 3-separators in time* $O(n^2)$.

Computing 2-Walks from Tutte paths and cycles [60]. It was shown by Gao, Richter, and Yu [30, 31] that in order to find a closed 2-walk in a circuit graph, it suffices to find a Tutte path that has a system of distinct representatives. We briefly recall the argument of [30, 31] below.

According to Lemma 2.14, $G \setminus x$ is a plane chain of blocks. By computing a Tutte path for every such block and extending the union of these Tutte paths to x (using the two incident edges in C), we immediately obtain a Tutte cycle of G (as in Corollary 2.4). Note that the time for computing this Tutte cycle is dominated by the computation of the Tutte paths.

To compute a closed 2-walk we will use the vertices of the SDR S as branch vertices at which the walk deviates from T into 2-walks of the T-bridges, which exist by induction. The constructed closed 2-walk will, therefore, have special properties for the vertices that are visited twice. Let an *internal 3-separator* S of a circuit graph (G, C_G) be a 3-separator such that $G - S$ contains a component disjoint from C.

Let T be a Tutte cycle and S be an SDR as given in Corollary 2.4. If G is a triangle, T is itself the desired 2-walk W of Theorem 2.5; otherwise, we use induction on the number m of edges in G. For every T-bridge L of G and its representative s in S, we consider a plane chain of blocks as follows.

If L has exactly two attachments (thus, L contains an edge of C_G), let t be the attachment different from s. Then $\{s, t\}$ is a 2-separator of G and $L \cup st$ is a circuit graph, according to Lemmas 2.11 and 2.12. According to Lemma 2.14, $(L \cup st) \setminus t$, and therefore, also $L \setminus t$ is a plane chain of blocks $B_1, \dots, B_l$ such that $s \in B_1$ and $t' \in B_l$ for the neighbor t' of t in $C \cap L$. Set $v_0 := v$ and $v_l := t'$.

If L has exactly three attachments $\{s, t, z\}$, $L \cup \{st, tz, zs\}$ is a circuit graph due to the 3-Path Property. By Lemma 2.15, $L \cup \{st, tz, zs\} \setminus \{t, z\} = L \setminus \{t, z\}$ is a plane chain of blocks $B_1, \dots, B_l$ such that $s \in B_1$ and $z' \in B_l$ for the neighbor z' of z on the boundary of L in direction s. Set $v_0 := s$ and $v_l := z'$.

Let v_i be the 1-separator $B_i \cap B_{i-1}$ of the constructed plane chain of blocks for every i. Each B_i is either an edge or a circuit graph. If B_i is an edge, we define an artificial walk $v_{i-1}, v_{i-1}v_i, v_i, v_iv_{i-1}, v_{i-1}$ for B_i; otherwise, there is a 2-walk in B_i by induction with $x := v_{i-1}$ and $y := v_i$. In both cases, v_i is visited exactly once, implying that the union W_L of these walks is a 2-walk of the plane chain of blocks, in which v is visited exactly once. Finally, we obtain the desired 2-walk W by traversing T from one representative s of a T-bridge to the next and detouring into W_L every time. Note that every s is visited exactly twice, once by T and once by W_L, as it is a representative in S.

For all steps taken in the description above, except for the computation of Tutte paths and the computation of suitable circuit subgraphs (i.e., the above plane chains of blocks) for the recursion on L, the corresponding existence proofs give immediately linear time algorithms.

We next show that a polynomial time computation of a Tutte path implies a polynomial time computation of a 2-walk. Assume that a Tutte cycle T of G and its SDR S can be computed in time cm^k for some integers c and k. If the 2-walks in the T-bridges have already been computed by recursion, taking the union of T and these 2-walks needs only linear time. Let $time(m)$ denote the running time of the resulting algorithm. We number all blocks of the plane chains of blocks that were constructed for T-bridges in G from 1 to j. Let m_i denote the number of edges in block i. As all these blocks are edge-disjoint and T contains at least one edge, $\sum_{i=1}^{j} m_i < m$. Thus, $time(m) = cm^k + \sum_{i=1}^{j} time(m_i) \leq cm^{k+1}$, as we always recurse on strictly smaller subgraphs and the recursion depth is at most m. Therefore, a proof of Theorem 2.6 as given in the following chapter implies Theorem 2.7.

CHAPTER 3

Computing Tutte Paths in Circuit Graphs

We will prove Theorem 2.3 by extending the decomposition of Gao, Richter, and Yu. The extended decomposition will only branch into edge-disjoint circuit graphs and thus turn out to be algorithmically accessible. In the following sections, we will first review some steps given in [30, 31] needed to set up the decomposition, then explain how we can avoid overlapping subgraphs, and finally give the details of the extended decomposition.

3.1 Setting up the Decomposition

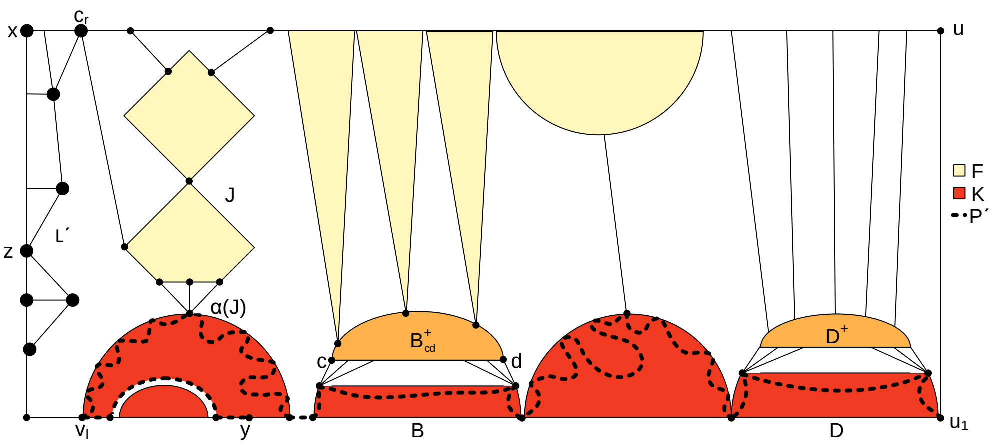


Figure 3.1: A circuit graph (G, C_G), in which the plane chain of blocks K is depicted in dark gray (red) and gray (orange), and F is the subgraph induced by xC_Gu and the vertices of light grey (yellow) and gray subgraphs. Here, F and K overlap in the gray subgraphs B^+ and D^+. The part P' from u_1 to y of the desired Tutte path of G can be computed by induction on the blocks of K.

We review the initial steps taken for the original decomposition in [30, 31]. Let (G, C_G) be a circuit graph, let $x, u, y \in V(C_G)$ with $x \neq y$ and let $a \in \{x, u\}$. We want to find a Tutte path from x to y through u. The vertex a acts as a place-holder that allows us to prevent x or u to be in the SDR S; this will be useful for the induction. We first eliminate some symmetric cases. If $u = x$, we can choose any other vertex $v \in V(C_G) \setminus x$ and assign $u = v$. The same holds if $u = y$ and $a \neq u$. If $a = u = y$, we interchange the roles of x and y and proceed as above. Thus we can assume that $u \notin \{x, y\}$. We will need y to be in uC_Gx in a later step. Therefore if $y \in xC_Gu$, we flip the current embedding of G such that in the new embedding $y \in uC_Gx$.

The proof of Theorem 2.3 proceeds by induction on the number of edges in G. If $|E(G)| = 3$, G is a triangle. In that case, the Tutte path we are looking for is xuy, the corresponding SDR S is empty. For the induction step, let u_1 be the neighbor of u in uC_Gx. In the special case that $u_1 = y$, we define $K := u_1$. Otherwise, we define K as the *minimal connected union* of blocks of $G \setminus xC_Gu$ that contains u_1 and y, where minimality is with respect to the number of blocks (see Figure 3.1). The blocks of K form a tree; by minimality, K will be a plane chain of blocks. Let $B_1, \ldots, B_l$ be the blocks of K such that $u_1 \in B_1$ and $y \in B_l$ and let C_{B_i} be the external face boundary of B_i. We number the 1-separators in K from v_1 to v_{l-1}, i.e., the blocks B_i and B_{i+1} intersect exactly in v_i. In addition, we set $v_0 := u_1$ and define v_l as the vertex in B_l nearest to x in u_1C_Gx. For simplicity, we divide the external face boundary C_{B_i} of any block B_i of K into its *lower part*, which is $v_{i-1}C_{B_i}v_i$, and its *upper part*, which is $v_iC_{B_i}v_{i-1}$. The *lower boundary of* K is then the union of the lower parts of all blocks of K, and the *upper boundary of* K is the union of the upper parts of all block of K.

3.2 Avoiding Overlapping Subgraphs

In the original proof of Theorem 2.3 given in [30, 31], the authors define a second connected subgraph F that overlaps with K and then recurse on both subgraphs separately by constructing Tutte paths of every block of these subgraphs (see Figure 3.1). The recursively constructed Tutte paths of F (giving a path from x to u) and in K (giving a path from u_1 to y) are then concatenated with uu_1 to get the desired Tutte path of G. The overlapping parts of F and K may, therefore, receive multiple recursive calls, which prevents to bound the running time of this decomposition. However, the description of F in [30, 31] suggests that an overlapping subgraph in this decomposition consists always of the inner vertex set of some bridge of the Tutte path computed for K. In the following, we will compute a Tutte path from u_1 to y, but instead of doing this in K, we will do this in a slightly modified subgraph $\eta(K)$. This augmentation will allow us to identify and exclude possible overlapping subgraphs in advance.

Contrast to the approach of [30, 31]: We explain the idea for our decomposition; the precise decomposition will be given in the next section. Let T be a Tutte path from u_1 to y of K and consider any T-bridge J of K. In the decomposition of [30, 31], by planarity, J can only take part in an overlapping if it intersects the upper external face boundary of K. Then J has exactly two attachments c and d, according to the definition of a Tutte path and the fact that J contains a boundary edge of some block of K. By Observation 2.9 and Lemma 2.11, c and d must be as well on the boundary of K. In fact, c and d are on the upper boundary of K by Lemma 2.16. In summary, the only parts of K that would have possibly overlapped in the original decomposition are the T-bridges with exactly two attachments on the upper boundary of K (drawn in gray (orange) in Figure 3.1). Thus, if we find for some block B_i of K all 2-separators in $v_iC_{B_i}v_{i-1}$ before we compute a Tutte path of this block, we have identified all subgraphs of this block which would have possibly overlapped in the original decomposition.

Now we give the details of this approach, which itself is a refined version of the proof given in [60]. Let $\{c, d\}$ be a 2-separator of a block B_i such that c and d are in $v_iC_{B_i}v_{i-1}$

(here we denote the vertex that appears first in $v_iC_{B_i}v_{i-1}$ by c and the other by d). Let further B^+_{cd} be the $\{c,d\}$-bridge in B_i that contains the path $cC_{B_i}d$ (see Figure 3.1). We call a 2-separator $\{c,d\}$ in $v_iC_{B_i}v_{i-1}$ *maximal* in $v_iC_{B_i}v_{i-1}$ if there is no other 2-separator $\{c',d'\}$ in $v_iC_{B_i}v_{i-1}$ with c and d in $c'C_{B_i}d'$. Note that in the special case $v_iv_{i-1} \in C_{B_i}$ two maximal 2-separators $\{v_i,c'\}$ and $\{d',v_{i-1}\}$ may occur that *interlace*, i.e. for which $v_iC_{B_i}c' \cap d'C_{B_i}c_{i-1} \neq \emptyset$. This is the only case in which two maximal 2-separators can interlace, since if otherwise $v_iv_{i-1} \notin C_{B_i}$, $\{v_i,v_{i-1}\}$ would be a 2-separator of B_i such that $v_iC_{B_i}v_{i-1}$ would contain both $v_iC_{B_i}c'$ and $d'C_{B_i}v_{i-1}$, which contradicts their maximality. We resolve this special case of having two interlacing maximal 2-separators by always using the one of these two that contains v_i in the following description and ignoring the other. Because of this, the maximal 2-separators taken for every block B_i will be consecutive on $v_iC_{B_i}v_{i-1}$. For the computation of a Tutte path of B_i, we will first find all maximal 2-separators in C_{B_i}. Possible smaller 2-separators inside them will only be computed if necessary.

Let $\{c,d\}$ be a 2-separator of B_i with c and d in $v_iC_{B_i}v_{i-1}$ and let v be an inner vertex of B^+_{cd}. Then c_l and c_r are defined as the vertices in xC_Gu closest to x and u, respectively, that are reachable from v in G by a path not containing any vertex of $\{c,d\} \cup V(C_G)$ as inner vertex (possibly $c_l = c_r$). Figure 3.3 shows two examples where $c_l \neq c_r$. For a 2-separator $\{c,d\}$ of B_i with c and d in $v_iC_{B_i}v_{i-1}$, let F'_{cd} be the $\{c,d,c_l,c_r\}$-bridge that contains B^+_{cd} and let $F_{cd} := F'_{cd} \setminus \{c,d\}$. (Continuing the above contrast to [30, 31], the graph F_{cd} contains the possibly overlapping parts of K of the original decomposition.)

In order to modify K to $\eta(K)$, we iterate through all maximal 2-separators $\{c,d\}$ of every block of K and "cut off" some B^+_{cd} in a predefined way. This will allow us to compute Tutte paths for every block of $\eta(K)$ and iteratively detour these Tutte paths to the subgraphs B^+_{cd} if necessary. For some B^+_{cd}, we will add a special edge to $\eta(K)$ whose containment in the previously computed Tutte path will decide whether such a detour is needed. The exact definition of $\eta(K)$ is dependent on the existence of a 1-separator in F_{cd}. For the relevant case $c_l \neq c_r$, we will prove that a vertex b is a 1-separator of F_{cd} if and only if $\{b,c,d\}$ is a 3-separator of G (see Figure 3.2). If such a 1-separator b exists, we will show that b can actually be chosen in such a way that the subgraph of F_{cd} "above" b is a block; such a vertex will additionally be unique.

Lemma 3.1. *Let $c_l \neq c_r$. A vertex $b \in F_{cd}$ is a 1-separator of F_{cd} if and only if $\{b,c,d\}$ is a 3-separator of G. No 1-separator of F_{cd} is contained in $c_lC_Gc_r$.*

Proof. Let b be any 1-separator of F_{cd}. We first show that $b \notin c_lC_Gc_r$, giving the second claim. Let J be the $c_lC_Gc_r$-bridge of F_{cd} containing the connected graph $B^+_{cd} \setminus \{c,d\}$. By definition of c_l and c_r, J contains c_l and $c_r \neq c_l$ as attachments. Every other $c_lC_Gc_r$-bridge in F_{cd} does not touch K and therefore has at least three attachments on $c_lC_Gc_r$ by the 3-Paths Property. Since $c_lC_Gc_r$ is a path, deleting any vertex of $c_lC_Gc_r$ in F_{cd} leaves a connected graph.

Consider any component of $F_{cd} \setminus b$ that does not contain $c_lC_Gc_r$. This component can have at most the neighbors $\{b,c,d\}$ in G. Since the component does not contain any vertex of C_G, its neighbor set in G must be exactly $\{b,c,d\}$, according to the 3-Paths Property. Thus, $\{b,c,d\}$ is a 3-separator of G.

Let $\{b,c,d\}$ be a 3-separator of G. Then $b \notin c_lC_Gc_r$, as otherwise $G \setminus \{b,c,d\}$ would be connected by definition of c_l and $c_r \neq c_l$. Consider any component of $F_{cd} \setminus b$ that does

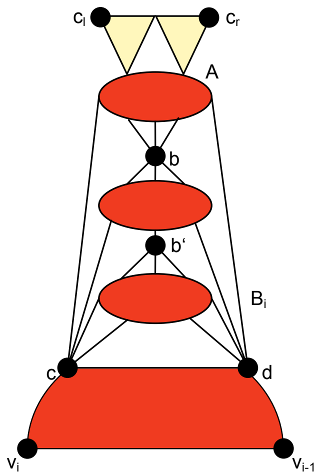


Figure 3.2: Two 1-separators b and b' of F_{cd}. The 1-separator b is the unique one contained in A.

not contain $c_l C_G c_r$. Since this component contains no vertex of C_G, its neighbor set in G is exactly $\{b, c, d\}$. Thus, b separates some vertex of that component from $c_l C_G c_r$ in F_{cd} and is therefore a 1-separator of F_{cd}. □

Lemma 3.1 implies that there is a block of F_{cd} that contains $c_l C_G c_r$. We call this block A. Note that there may be many 1-separators in F_{cd} (see Figure 3.2). However, there is exactly one such 1-separator that is contained in A.

Lemma 3.2. *Let $c_l \neq c_r$ and let F_{cd} contain a 1-separator. Then F_{cd} contains a unique 1-separator b such that $b \in A$.*

Proof. Since F_{cd} has a 1-separator and by the maximality of the block A of F_{cd}, A contains at least one 1-separator b of F_{cd}. Assume to the contrary that A contains a 1-separator $b' \neq b$ of F_{cd}. Let H_1 and H_2 be components of $F_{cd} - b$ and $F_{cd} - b'$, respectively, that do not contain $c_l C_G c_r$. As 1-separators that are contained in the same block A separate disjoint components from A (as implied by the block-cut-tree), H_1 and H_2 are disjoint; moreover, both do not contain any vertex of C_G. By the 3-Paths Property, H_1 and H_2 are neighbored exactly to $\{b, c, d\}$ and $\{b', c, d\}$ in G, respectively. Then the union of C_G and the set of three paths from H_1 and from H_2 to C_G due to the 3-Paths Property form a $K_{3,3}$, which contradicts the planarity of G. □

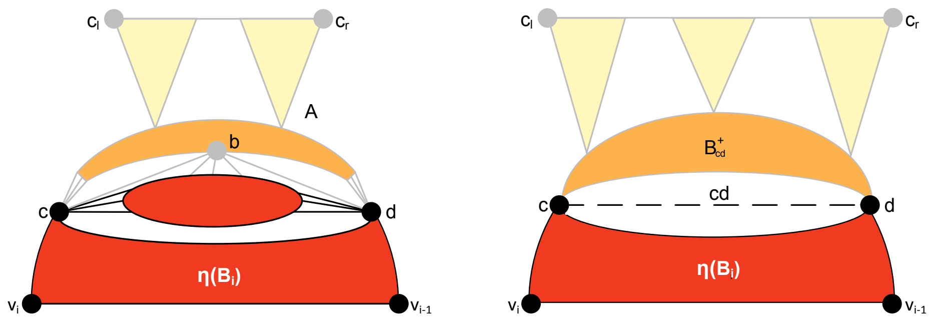
In the following, whenever dealing with a maximal 2-separator $\{c, d\}$ of K, the variables $F_{cd}, F'_{cd}, c_l, c_r, B_i, A$ will always refer to the previously defined objects and b will refer to the unique 1-separator of F_{cd} defined in Lemma 3.2. We are now ready to define $\eta(K)$.

Definition 3.3. *Let $\eta(K)$ be the graph obtained from K by performing the following for every maximal 2-separator $\{c, d\} \neq \{v_i, v_{i-1}\}$ of every block B_i of K.*

Case 1*: $c_l = c_r$*
Do nothing.

Case 2: *$c_l \neq c_r$ and F_{cd} contains a 1-separator (see Figure 3.3(a))*
Replace B_{cd}^{+} with $B_{cd}^{+} \setminus A$.

Case 3: *$c_l \neq c_r$ and F_{cd} contains no 1-separator (see Figure 3.3(b))*
Delete all inner vertices of B_{cd}^{+} and add the edge cd if cd does not already exist.



(a) Case 2: $c_l \neq c_r$ and F_{cd} contains a 1-separator b. We replace B_{cd}^{+} with $B_{cd}^{+} \setminus A$.

(b) Case 3: $c_l \neq c_r$ and F_{cd} does not contain a 1-separator. We delete all inner vertices of B_{cd}^{+} and add the edge cd if it does not already exist.

Figure 3.3: The two cases of modifying K to $\eta(K)$. In both cases, the remaining part of B_{cd}^{+} is the dark gray (red) subgraph, i.e., the gray (orange) part of B_{cd}^{+} is deleted.

For a block B_i of K, let $\eta(B_i)$ be the corresponding block of $\eta(K)$. Let $\eta(C_{B_i})$ be the external boundary of $\eta(B_i)$. Note that $\eta(K)$ is no longer a plane chain of blocks of $G \setminus xC_Gu$, as the modified blocks $\eta(B_i)$ are no longer maximal in G. However, every $\eta(B_i)$ that is not just an edge is still a circuit graph, as shown next.

Lemma 3.4. *Every $\eta(B_i)$ that is not an edge is a circuit graph.*

Proof. Clearly the claim is true when $\eta(B_i) = B_i$, thus assume the contrary. We consider a B_i after one Case 2 or Case 3 modification of Definition 3.3; the arguments extend readily to multiple such modifications.

Consider a Case 2 modification. Let b be the unique 1-separator of Lemma 3.2. According to Lemma 3.1, $\{b, c, d\}$ is a 3-separator of G. Let H denote the (unique) $\{b, c, d\}$-bridge of G that does not contain a vertex of C_G. By the definition of bridges, $H \setminus b$ is connected. We show that the boundary part of $H \setminus b$ from C_G to d that contains all former neighbors of b is a path. Otherwise, an clockwise boundary traversal from C_G to d would visit some vertex z twice, which gives a 2-separator $\{z, b\}$ that contradicts Lemma 2.11. Thus, the claim follows directly from extending this path by $dC_{B_i}c$ (which is internally disjoint) and applying Lemma 2.13.

Consider a Case 3 modification. Then $B_{cd}^{+} \cup cd$ is a circuit graph by Lemma 2.12. □

3.3 Extending the Decomposition

We extend the decomposition described so far. From now on, we will name the input graph $(G', C_{G'})$ instead of (G, C_G), but keep all other notation such as K, B_i, x, y, u, u_1.

For every $(K \cup xC_{G'}u)$-bridge L in G', L intersects K in at most one vertex, as otherwise, a block of K would not be maximal. We call this vertex, if it exists, $\alpha(L)$. Note that the edge uu_1 is not a $(K \cup xC_{G'}u)$-bridge by definition. It is however possible that there is a $(K \cup xC_{G'}u)$-bridge that contains $v_lC_{G'}x$. If so, we denote this special bridge by L' (otherwise, $v_lC_{G'}x$ is just an edge). The bridge L' is special among the $(K \cup xC_{G'}u)$-bridges, as it is the only one that may have exactly two attachments; all other bridges have at least three attachments by the 3-Path Property.

For a $(K \cup xC_{G'}u)$-bridge L, let $C_{G'}(L)$ be the shortest path in $v_lC_{G'}u$ that contains all attachments of L in $v_lC_{G'}u$. When considering such L, the endvertices of $C_{G'}(L)$ closest to v_l and u in $v_lC_{G'}u$ are called c_l and c_r, respectively ($c_l = c_r$ is possible). For such L, let $J(L)$ denote the $\{c_l, c_r\}$-bridge of G' that contains L.

A $(K \cup xC_{G'}u)$-bridge $L \neq L'$ in G' is *isolated* if $\alpha(L)$ does not exist (i.e., $L \cap K = \emptyset$), and its 2-separator $\{c_l, c_r\}$ of $xC_{G'}u$ is maximal in $xC_{G'}u$ with respect to the 2-separators of all other such bridges. Thus, L is different from L' and has at least three attachments on $xC_{G'}u$.

We now transform the input graph $(G', C_{G'})$ into a graph (G, C_G) that does not contain L' anymore. If L' does not exist in G', then we simply set $G := G'$. Otherwise, we apply the following modification on G' that depends on the number of attachments of L'. If L' has exactly two attachments (namely, v_l and x), obtain G from G' by replacing L' with the edge v_lx. Otherwise, L' has at least the three attachments v_l, x, c_r (as is the case in Figure 3.1). Let $L^* := (L' \cup C_{G'}(L')) \setminus v_l$. Note that L^* may not be 2-connected. We obtain G from G' by contracting L^* to one vertex c_r (which will be x in G) and subsequently deleting multiedges.

Note that K and $\eta(K)$ are the same for G and G'. In Section 3.3, we will find a Tutte path of $\eta(K)$ and an SDR S of its bridges. In Section 3.3, this Tutte path of $\eta(K)$ will be modified to a Tutte path of G. Eventually, we show in Section 3.3 how to deal with the special bridge L' in G' and thereby extend the Tutte path and its SDR found in G to a Tutte path of G'.

Finding a Tutte Path of $\eta(K)$

We continue the decomposition of the circuit graph (G, C_G) (as described in Section 3.1) by computing a Tutte path $P_{\eta(K)}$ of $\eta(K)$ from u_1 to y and an SDR $S_{\eta(K)}$ of the $P_{\eta(K)}$-bridges. For each block $\eta(B_i)$ of $\eta(K)$, we compute $P_{\eta(B_i)}$ and an SDR $S_{\eta(B_i)}$ of the $P_{\eta(B_i)}$-bridges as follows.

If $\eta(B_i)$ is just an edge $v_{i-1}v_i$, set $P_{\eta(B_i)} := v_{i-1}v_i$ and $S_{\eta(B_i)} := \emptyset$. Otherwise, if $i < l$, compute by induction a Tutte path $P_{\eta(B_i)}$ of $\eta(B_i)$ from v_{i-1} to v_i and an SDR $S_{\eta(B_i)}$ of all $P_{\eta(B_i)}$-bridges such that $v_i \notin S_{\eta(B_i)}$ (as intermediate vertex, an arbitrary vertex in $V(C_{B_i}) \setminus \{v_{i-1}, v_i\}$ can be chosen). If $i = l$, compute a Tutte path $P_{\eta(B_l)}$ of $\eta(B_l)$ from v_{l-1} to y through v_l and an SDR $S_{\eta(B_l)}$ of all $P_{\eta(B_l)}$-bridges. Since we may need $v_l \in B_l$ as representative for L' in Section 3.3, we have to ensure that v_l does not become a representative for any $P_{\eta(B_l)}$-bridge in $\eta(B_l)$. Thus, apply the induction on $\eta(B_l)$ such that $v_l \notin S_{\eta(B_l)}$. Then $P_{\eta(K)} = \cup_{i=1}^{l} P_{\eta(B_i)}$ is the desired Tutte path of $\eta(K)$ from u_1 to y and $S_{\eta(K)} = \cup_{i=1}^{l} S_{\eta(B_i)}$ is an SDR of $P_{\eta(K)}$s bridges in $\eta(K)$.

Every $P_{\eta(B_i)}$-bridge with three attachments in $\eta(B_i)$ is also a $P_{\eta(B_i)}$-bridge with three attachments in G. Every internal vertex of such a $P_{\eta(B_i)}$-bridge has the same neighborhood in $\eta(B_i)$ as in G. Therefore, each such bridge preserves its number of attachments in G. The same argument holds for the $P_{\eta(B_i)}$-bridges in $\eta(B_i)$ that have exactly two attachments and contain an edge of C_G. In fact, these two observations do not only hold for $P_{\eta(B_i)}$, but for any Tutte path P_H of some circuit graph $H \subset G$. We will therefore only discuss P_H-bridges in the remainder of this thesis, that have exactly two attachments in H and do not contain any edge of C_G. We will show that these bridges have exactly three attachments in G.

Finding a Tutte Path of G

In order to find the desired Tutte path P of (G, C_G) and an SDR S for its bridges, we initially set $P := xC_Gu_1 \cup P_{\eta(K)}$ and $S := S_{\eta(K)}$, and then modify P and S step by step such that the final path P is a Tutte path of (G, C_G), does not contain any edge cd that was added in Case 3 of the definition of η, and S is an SDR of all P-bridges. We will decompose G into smaller circuit graphs on which we apply induction. These graphs will pairwise intersect in at most one vertex, i.e., they are *edge-disjoint*. By carefully choosing a when applying the induction, we will ensure that the intersection vertex is a representative in at most one intersecting graph. The modification of P starts by handling the $(K \cup xC_Gu)$-bridges that have an attachment on K, but are not contained in any F_{cd}. We next show useful details of these bridges.

Let L be any $(K \cup xC_Gu)$-bridge for which $\alpha(L)$ exists and which is not contained in some F_{cd}.

Lemma 3.5. *$\alpha(L) \in \eta(K)$ and $\alpha(L) \in P_{\eta(B_i)}$.*

Proof. For the first claim, assume to the contrary that $\alpha(L) \in K$ is not in $\eta(K)$. Then $\alpha(L)$ lies on the boundary of K on a path between the vertices of a maximal 2-separator $\{c, d\}$ and thus must be part of F_{cd}, contradicting the assumption.

Next, we assume $\alpha(L) \notin P_{\eta(B_i)}$. As $\alpha(L)$ is on the boundary of $\eta(B_i)$, $\alpha(L)$ must be contained in a $P_{\eta(B_i)}$-bridge in $\eta(B_i)$ with two attachments $\{c', d'\}$. By Observation 2.9, $\{c', d'\}$ is a 2-separator of $\eta(B_i)$. As L is not contained in some F_{cd}, $\{c', d'\}$ is not a maximal 2-separator of B_i. Thus, there exists a maximal 2-separator $\{c, d\}$ with $c'C_{B_i}d' \subseteq cC_{B_i}d$. This gives a contradiction, as then by construction $\alpha(L) \notin \eta(K)$. □

Let J' be the union of $L, C_G(L)$ and all $C_G(L)$-bridges of G which have all their attachments in $C_G(L)$. Let $J = J' \setminus \alpha(L)$.

Lemma 3.6. *J is a circuit graph.*

Proof. We first prove that J is 2-connected: L has an inner vertex by the definition of a bridge and thus at least two attachments on C_G by the 3-Paths Property. Hence, $|V(J)| \geq 3$. Starting with $C_G(L)$ and adding the two paths to $C_G(L)$ from every remaining vertex in J due to the 3-Paths Property gives an *open ear decomposition* [75]. Thus, J is 2-connected. It follows that the boundary of J is a cycle and J is a circuit graph. □

We are now ready to describe an algorithm that, given the circuit graph (G, C_G), vertices $x, u, y \in V(C_G)$ and the preliminary Tutte path P as defined above, outputs a Tutte path of G and an SDR of its bridges in G.

Algorithm 1: *FindTuttePath*$((G, C_G), x, u, y, P, S)$

Input: $(G, C_G), x, u, y, P, S$, where P is the preliminary Tutte path $xC_Gu_1 \cup P_{\eta(K)}$ from x to y and $S = S_{\eta(K)}$ the corresponding SDR.

Output: A Tutte path of (G, C_G) and an SDR of its bridges in G stored in P and S respectively.

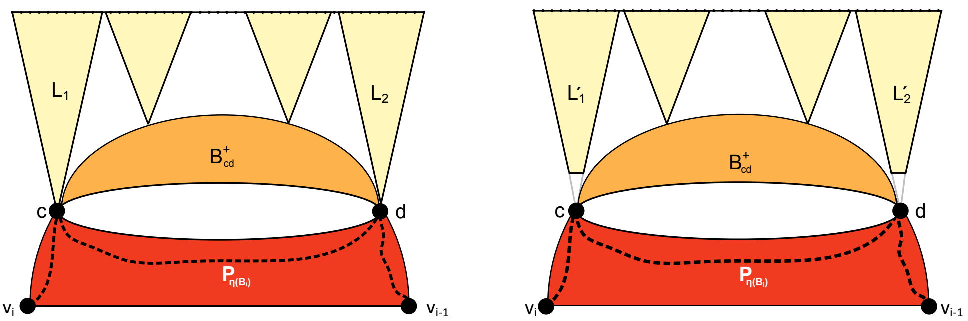
(1) For every $(K \cup xC_Gu)$-bridge L in G with $\alpha(L) \in \eta(K)$ (see Figure 3.4):

- According to Lemma 3.5, $\alpha(L) \in P_{\eta(B_i)}$ for some B_i.
- Let J' be the union of $L, C_G(L)$ and all $C_G(L)$-bridges of G which have all their attachments in $C_G(L)$. Let $J = J' \setminus \alpha(L)$. By Lemma 3.6, J is a circuit graph.

(a) Compute a Tutte path P_J from c_l to c_r and an SDR S_J of all P_J-bridges by induction such that depending on a, either c_l or c_r is not in S_J: if $a = x$, apply the induction such that $c_l \notin S_J$; otherwise, if $a = u$, apply the induction such that $c_r \notin S_J$.

(b) Set $P := P \setminus C_G(L) \cup P_J$ and $S := S \cup S_J$.

- By the 3-Paths Property, every P_J-bridge in J that has exactly two attachments and does not contain an edge of C_G must contain a vertex that in G is a neighbor of $\alpha(L)$. Each such P_J-bridge will therefore become a P-bridge with exactly three attachments in G.



(a) Only the $(K \cup xC_Gu)$-bridges L_1 and L_2 have an attachment in $\eta(K)$.

(b) From L_1 and L_2, circuit graphs L_1' and L_2' are constructed. We compute a Tutte path in each of them.

Figure 3.4: Step 1 of *FindTuttePath*. We consider the $(K \cup xC_Gu)$-bridges that have an attachment in $P_{\eta(B_i)}$ (dashed line).

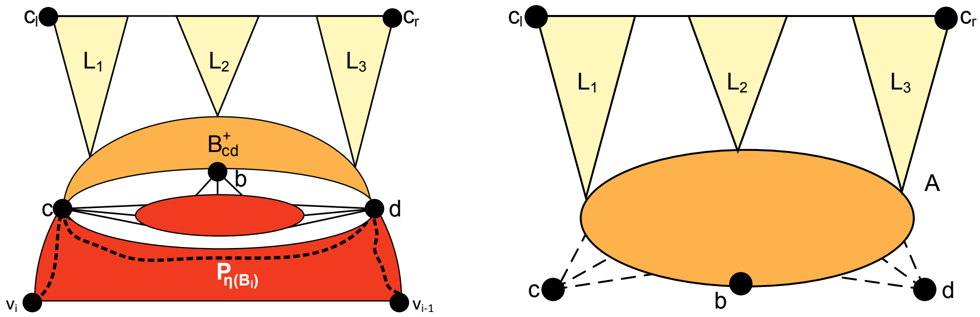
(2) For every maximal 2-separator $\{c, d\}$ of K satisfying Case 1 of Definition 3.3:

- Let J be any $P_{\eta(B_i)}$-bridge in $\eta(B_i)$ that contains an edge of $c\eta(C_{B_i})d$ (recall that $\eta(C_{B_i})$ denotes the external boundary of $\eta(B_i)$). We show that every such J becomes a P-bridge in G with exactly three attachments. By the 3-Path Property, there is a path from every inner vertex of J to some vertex in C_G that contains neither C_G nor d. In this case the only possible such vertex is

$c_l = c_r$. Thus, J is a P-bridge in G with exactly three attachments, one of which is c_l and its representative in S will be as chosen in $S_{\eta(B_i)}$.

(3) For every maximal 2-separator $\{c, d\}$ of K satisfying Case 2 of Definition 3.3 (see Figure 3.5):

(a) Compute a Tutte path P_A of the block A of F_{cd} from c_l to c_r through b and an SDR S_A of all P_A-bridges. If $a = x$, apply the induction such that $c_l \notin S_J$. Otherwise, if $a = u$, apply the induction such that $c_r \notin S_J$.



(a) A maximal 2-separator $\{c, d\}$ of B_i such that $c_l \neq c_r$ and F_{cd} contains a 1-separator. The unique 1-separator of F_{cd} in $A \subset F_{cd}$ is b.

(b) The block A of F_{cd} that contains $c_l C_G c_r$ (dashed edges are not part of A). We compute a Tutte path P_A of A from c_l to c_r through b.

Figure 3.5: Step 3 of *FindTuttePath*

(b) Set $P := P \setminus c_l C_G c_r \cup P_A$ and $S := S \cup S_A$.

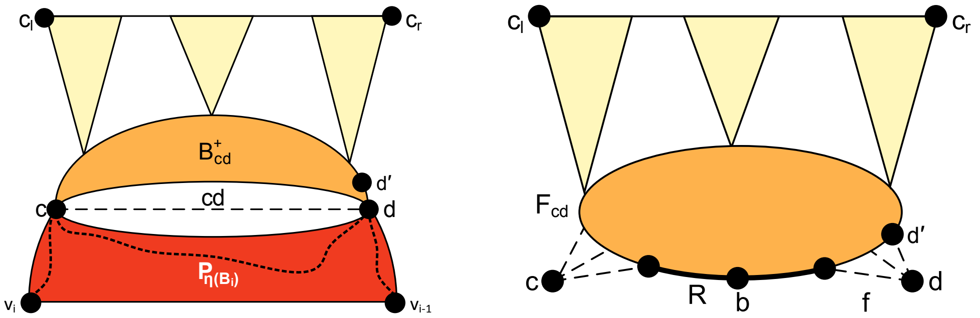
- Let H be the $\{b, c, d\}$-bridge in G that does not contain $c_l C_G c_r$, according to Lemma 3.1.
- Consider any P_A-bridge J with exactly two attachments in A that does not contain an edge of C_G. By the 3-Paths Property, J must contain an inner vertex that has a neighbor in $G \setminus A$. Since b is a 1-separator of F_{cd} in A and $b \in P_A$, the set of all such neighbors is either $\{c\}$, $\{d\}$ or $\{c, d\}$. We will show that the last case is not possible. Namely, as P_A is a Tutte path and J has only two attachments, J contains an edge of the external boundary of A. By planarity and the existence of (the connected) $\{b, c, d\}$-bridge H in G, J cannot be adjacent to both, C_G and d. Hence, every such P_A-bridge will become a P-bridge with exactly three attachments in G.
- In the case that $P_{\eta(B_i)}$ contains an edge of H, there may exist a $P_{\eta(B_i)}$-bridge $J \subseteq H \setminus b$ with two attachments having both attachments in $c\eta(C_{B_i})d$. By the 3-Path Property, there is a path from every inner vertex of J to some vertex in C_G that contains neither C_G nor d. As $J \subset H$, this path contains b. Thus, J is a P-bridge in G with exactly three attachments, one of which is b.
- By applying the induction depending on the value of a, we ensure that a is not a representative of any bridge in the final SDR S. Furthermore,

it ensures that the vertex in the intersection of two subgraphs F_{cd} and $F_{c'd'}$ is used as a representative in the result of at most one induction call made by the algorithm.

(4) For every maximal 2-separator $\{c, d\}$ of K satisfying Case 3 of Definition 3.3:

(a) If $cd \notin P_{\eta(B_i)}$ (see Figure 3.6):

- Let f be the face in B_i that contains cd and an inner vertex of B^+_{cd}.
- Let R be the path obtained from the boundary of B^+_{cd} in f by deleting C_G and d.



(a) A maximal 2-separator $\{c, d\}$ of B_i such that $c_l \neq c_r$ and F_{cd} contains no 1-separator. In this case, cd is not contained in $P_{\eta(B_i)}$.

(b) The subgraph F_{cd} (not containing dashed edges). We compute a Tutte path $P_{F_{cd}}$ of F_{cd} from c_l to c_r through $b \in R$ (the fat line depicts the path R).

Figure 3.6: Step 4(a) of *FindTuttePath*

i. Choose an arbitrary vertex b in R.

ii. Compute a Tutte path $P_{F_{cd}}$ of F_{cd} from c_l to c_r through b by induction on F_{cd} and an SDR $S_{F_{cd}}$ of all $P_{F_{cd}}$-bridges. If $a = x$, apply the induction such that $c_l \notin S_J$. Otherwise, if $a = u$, apply the induction such that $c_r \notin S_J$.

iii. Set $P := P \setminus c_l C_G c_r \cup P_{F_{cd}}$ and $S := S \cup S_{F_{cd}}$.

- Consider any $P_{F_{cd}}$-bridge J with exactly two attachments in F_{cd} that does not contain an edge of C_G. By the 3-Paths Property, the inner vertex set of J is neighbored to either $\{c\}$, $\{d\}$ or $\{c, d\}$. We show that the last case is not possible, which proves that every such $P_{F_{cd}}$-bridge becomes a P-bridge in G with exactly three attachments. By the choice of R, the only vertex that may be adjacent to C_G and d is b (in that case, $R = \{b\}$). However, b is not a neighbor of an inner vertex of J, as $b \in P_{F_{cd}}$. This proves the claim.

(b) If $cd \in P_{\eta(B_i)}$ (see Figure 3.7):

- Recall that cd was possibly added during the construction of $\eta(K)$ and may therefore not be in G. We aim to replace cd in $P_{\eta(B_i)}$ with a Tutte path of B^+_{cd} from C_G to d.

- This case is more complicated than the previous ones due to the fact that both C_G and d could already be representatives in S. The induction hypothesis allows us to protect only one vertex by choosing the parameter a. In the following, we will therefore apply induction on a modification of the graph $B^+_{cd} \cup F_{cd}$ such that d is not contained in this graph and $c \notin S$ in the end.
- According to Lemma 2.12, $B^+_{cd} \cup cd$ is a circuit graph.
- Let d' be the neighbor of d on the boundary of $B^+_{cd} \cup cd$ that is different from C_G.
- Let $K' := (B^+_{cd} \cup cd) \setminus d$. According to Lemma 2.14, K' is a plane chain of blocks $B'_1, B'_2, \dots, B'_{l'}$ such that $d' \in B'_1$ and $c \in B'_{l'}$. Note that K' is a subgraph of G, as it does not contain cd.
- By planarity, every $(K \cup xC_Gu)$-bridge L in G that is contained in F_{cd} has its attachment $\alpha(L)$ (if exists) on the upper boundary of K' (see Figure 3.7(b)), while every neighbor of d is on the lower boundary of K'.
- We will replace $cd \in P_{\eta(B_i)}$ with the union of the edge dd' and a Tutte path of $\eta(K')$ from d' to C_G; the Tutte path is constructed in the very same way as we did for K, i.e., by first computing $\eta(K')$, then Tutte paths of the blocks of $\eta(K')$ and then branching into the different steps of *FindTuttePath*. This will iterate on the maximal 2-separators of K', which are the sets of next smaller 2-separators of K. Note that $\eta(K)$ and $\eta(K')$ are edge-disjoint.
- Technically, $\eta()$ is defined on a given circuit graph. We face this problem by constructing the following artificial circuit graph $\overline{G}$, which allows for a proper definition of $\eta(K')$.
  - Let $\overline{G}$ be the union of $K' \cup c_lC_Gc_r$, all $(K \cup xC_Gu)$-bridges that are contained in F_{cd}, and the new edges cc_l and c_rd'. Clearly, $\overline{G}$ is a circuit graph $(\overline{G}, C_{\overline{G}})$. Let $x' := c_l$, $u' := c_r$, $u'_1 := d'$ and $y' := c$.
  - Then K' is consistent to our previous definition, i.e., the *minimal connected union* of blocks of $\overline{G} \setminus x'C_{\overline{G}}u'$ that contains y' and u'_1, and $\eta(K')$ is well-defined in dependence of $\overline{G}$ and $\{x', u', y'\}$.

i. Compute $\eta(K')$ from K'.

ii. For each block $\eta(B'_i)$ of $\eta(K')$, compute a Tutte path $P_{\eta(B'_i)}$ and an SDR $S_{\eta(B'_i)}$ of the $P_{\eta(B'_i)}$-bridges in $\eta(B'_i)$ by induction, as described in Section 3.3.

iii. Set $P' := c_lPc_r \cup P_{\eta(B'_1)} \cup \dots \cup P_{\eta(B'_{l'})} \cup c_rd'$.

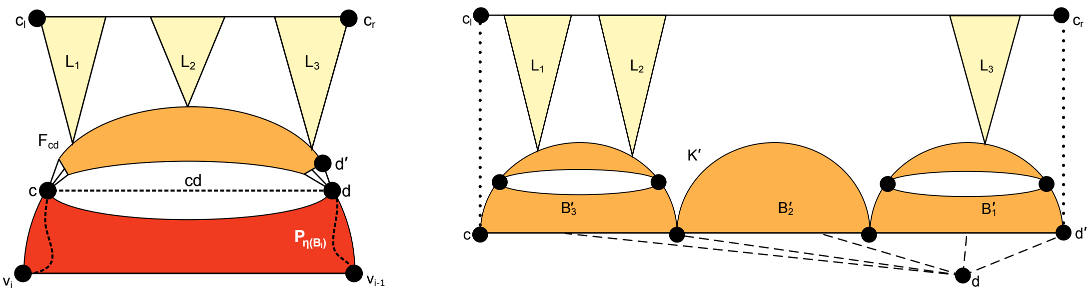
iv. Set $S' := S_{\eta(B'_1)} \cup \dots \cup S_{\eta(B'_{l'})}$.

v. Apply *FindTuttePath*$((\overline{G}, C_{\overline{G}})), x', u', y', P', S')$.

vi. Set $P := P \setminus c_lC_Gc_r \setminus cd \cup xPc_l \cup c_lP'c_r \cup c_rPd \cup dd' \cup d'P'c \cup cPy$.

vii. Set $S := S \cup S'$.

- By construction, $(\overline{G}, C_{\overline{G}})$ contains neither an L'-bridge nor an isolated bridge; moreover, P' is exactly the preliminary Tutte path of $(\overline{G}, C_{\overline{G}})$ computed in Section 3.3. Thus, *FindTuttePath*$((\overline{G}, C_{\overline{G}})), x', u', y', P')$



(a) A maximal 2-separator $\{c, d\}$ of B_i such that $c_l \neq c_r$ and F_{cd} contains no 1-separator. In this case, cd is contained in $P_{\eta(B_i)}$.

(b) The circuit graph $(G', C_{G'})$ (not containing dashed edges), which contains the plane chain of blocks K'. We iterate the computation of a Tutte path on $\eta(K')$ in $(G', C_{G'})$, which corresponds to iterating on the next smaller maximal 2-separators of K.

Figure 3.7: Step 4(b) of *FindTuttePath*

outputs a Tutte path of $(\overline{G}, C_{\overline{G}})$ and stores it in P'. The above construction of P then applies the changes that were made for P' to P.

- Since P' is a Tutte path of $(\overline{G}, C_{\overline{G}})$, the only P'-bridges with two attachments that do not contain an edge of C_G must have an inner vertex that is a neighbor of d by the 3-Paths Property. As $d \in P$, such P'-bridges will become P-bridges with exactly three attachments in G.

(5) For every isolated $(K \cup xC_Gu)$-bridge L in G:

- Any isolated bridge L that is contained in some F_{cd} for a maximal 2-separator $\{c, d\}$ of K has already been part of a recursive call that computed the Tutte-path $P_{F_{cd}}$. Therefore, it does not have to be considered again and we restrict ourselves to isolated bridges that are not contained in some F_{cd}.
- Any path from x to u in $G \setminus K$ must pass through $J(L)$ and, in particular, through the vertices $\{c_l, c_r\}$ of L. In G, P contains $c_l C_G c_r$. Recall that $J(L)$ is a circuit graph. We aim for replacing the subpath $c_l C_G c_r$ in P with a Tutte path of $J(L)$ from c_l to c_r.
- As c_l or c_r may already be in S, we have to be careful about how we apply the induction. In Steps 1(a), 3(a) and 4(a), we applied the induction on all F_{cd} subgraphs depending on the vertex a; hence, we know that not both c_l and c_r are already in S. In order to ensure that we do not add a to S in the case that $a \in J(L)$ (for example if $a = x$) and neither reuse c_l nor c_r as a representative, we will apply the induction in the same fashion depending on a.

(a) Compute a Tutte path $P_{J(L)}$ of $J(L)$ from c_l to c_r by induction on $J(L)$ and an SDR $S_{J(L)}$ of all $P_{J(L)}$-bridges. If $a = x$, apply the induction such that

$c_l \notin S_{J(L)}$. Otherwise, if $a = u$, apply the induction such that $c_r \notin S_{J(L)}$.

(b) Set $P := P \setminus c_l C_G c_r \cup P_{J(L)}$ and $S := S \cup S_{J(L)}$.

- Since L is an isolated bridge, every $P_{J(L)}$-bridge has no neighbor in $G \setminus J(L)$, and therefore does not change its number of attachments as a bridge of P in G.

Dealing with L'

We show how to deal with the bridge L' that we removed in advance. In the graph G, let P be a Tutte path from x to y through u with SDR S of all P-bridges in G, as computed by Algorithm 1. Assume that the bridge L' exists in G', as otherwise there is nothing to do. If L' has exactly two attachments in G' (namely x and v_l), then $v_l \in P$ and $v_l \notin S$ by the construction of P. In that case, P is a Tutte path of G', L' is a P-bridge of G' with two attachments, and we simply add v_l to S as the representative of L'.

Otherwise, L' has at least the three attachments v_l, x, c_r in G' (see Figure 3.1). Let $L^* := (L' \cup C_G(L')) \setminus v_l$. Note that L^* may not be 2-connected. The following steps will extend the subpath $c_r P y$ of P and the SDR S computed by Algorithm 1 to a Tutte path from x to y and SDR of G'.

(1) If L^* is not 2-connected:

- Every 1-separator z of L^* is contained in $v_l C_G c_r \setminus v_l$ (note that $v_l \notin L^*$), as otherwise $\{z, v_l\}$ would be a 2-separator with $z \notin C_G$, contradicting Lemma 2.11. Furthermore, $z \in v_l C_G x \setminus v_l$, as otherwise $z \in x C_G c_r \setminus x$ would imply that $\{z, v_l\}$ is a 2-separator that violates the choice of L' (e.g., c_r would not be an attachment of L').

(a) Let z be the 1-separator of L^* in $v_l C_G x \setminus v_l$ closest to x (possibly $z = x$).

(b) Let L^*_B be the $\{z\}$-bridge of L^* containing c_r.

(2) If L^* is 2-connected:

(a) Let z be the neighbor of v_l in $C_G \cap L^*$.

(b) Let $L^*_B := L^*$.

(3) Compute a Tutte path $P_{L^*_B}$ of L^*_B from x to c_r through z and an SDR $S_{L^*_B}$ of all $P_{L^*_B}$-bridges in L^*_B by induction.

- In both cases, L^*_B is 2-connected and thus a circuit graph due to Lemma 2.13.
- We apply the induction such that, depending on the value of a, either x or c_r is not in S_{L^*}.

(4) Obtain the Tutte path $P := P_{L^*_B} \cup c_r P y$ of G' with SDR $S := S_{L^*_B} \cup S$.

- If L^* is not 2-connected, then the part of L^* that is not contained in L^*_B becomes a P-bridge with attachments v_l and z. As Algorithm 1 ensures $v_l \notin S$, we can add v_l to S as the representative of that P-bridge.

3.4 A Quadratic Time Bound

We consider the overall algorithm A on the circuit graph G' with m edges, which modifies G' to a graph G having no special bridge L' (see beginning of Section 3.3), then computes a preliminary Tutte path P of G (see Section 3.3), and eventually invokes Algorithm 1 to extend P. Let $time(m)$ be the running time of Algorithm A on G'. We need to show that $time(m) = O(m^2) = O(n^2)$.

Clearly, all recursive calls of Algorithm A are made on pairwise edge-disjoint circuit graphs. For detecting blocks and chains of blocks, we use any algorithm such as [64] that is able to compute the 2-connected components of a graph in linear time. Every single step of Algorithm A that is not a recursive call uses elementary graph operations or computes maximal 2-separators and can, therefore, be done in time $O(m)$.

It thus suffices to show that the number of recursive calls of A is linear in m and that we did not add too many new edges for every recursive call. If j recursive calls were invoked on G', let G_i be the circuit graph of the ith such call and let $m_i := |E(G_i)|$ for all $1 \leq i \leq j$.

If we would not add any new edge during the computation of A, every G_i would be a subgraph of G' and we would have the recurrence $time(m) = O(m) + \sum_{i=1}^{j} time(m_i)$. Let w be the neighbor of v_l in $v_l C_G x$. As all G_i are edge-disjoint and do not contain the edges uu_1 and $v_l w$, we have $\sum_{i=1}^{j} m_i \leq m - 2$. Solving the recurrence above gives then $time(m) = O(m^{1+1}) = O(n^2)$.

However, we may have added a new edge cd in Algorithm A only when constructing $\eta(K)$ (in Case 3 of Definition 3.3), either during the computation of the preliminary Tutte path P or before the recursive call of Algorithm 1 in Case 4(b)i. Every such new edge cd is part of exactly one recursive call made for G' on a graph G_i that is a block of ηK (see Section 3.3). However, for every such G_i and cd, the unique edge dd' (as shown in Figure 3.6(a)) is not contained in any recursive call made for G'. Since this edge dd' compensates the additional edge cd, this restores the validity of the above argument.

This proves Theorem 2.6 and hence Theorem 2.7. The most crucial open question is how the given cubic running time for computing a special closed 2-walk can be improved to a polynomial of lower order.

CHAPTER 4

Tutte Paths in 2-Connected Planar Graphs

In this chapter, we shift our focus from circuit graphs to all 2-connected plane graphs. For this, we broadly follow the idea of [18] and construct a Tutte path that is based on the appearance of certain 2-separators in the graphs constructed during our decomposition of the given graph. This depends on many structural properties of the input. In [18], the necessary properties to compute a Tutte path in linear time follow from the restriction to the class of internally 4-connected planar graphs, the restriction on the endvertices of the desired Tutte path, and the fact that the Tutte paths computed recursively are actually Hamiltonian paths. In contrast, here we give new insights into the much wider structural variety of Tutte paths of 2-connected planar graphs. In addition, as stated in Theorem 2.8, we allow $x, y \notin C_G$, and hence extend the techniques used in [18]. We show that based on the prescribed vertices and edge, there is always a set of unique 2-separators that must be contained in any Tutte path of the given graph. We then use this set of 2-separators to iteratively construct a preliminary Tutte path and apply this iterative procedure such that we avoid overlappings of more than one edge while decomposing the input graph. Other than in the previous chapter we will not be able to compute an SDR for the constructed Tutte path, as such a system does not necessarily exist for every Tutte paths in 2-connected planar graphs (Figure 4.1 shows a simple example where this is the case).

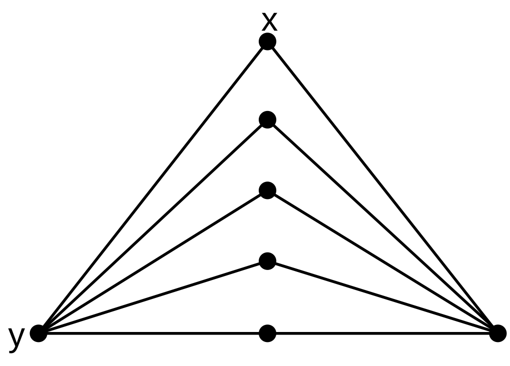


Figure 4.1: An example of a 2-connected plane graph where there is no SDR for the bridges of any Tutte cycle through the vertices x and y. Any such cycle would have at least three bridges with the same two attachments.

We start by excluding two instances for which it is easy to show that Theorem 2.2 holds. With these two easy cases out of the way, we assume that our input instance does not fall into one of them and show how this assumption allows us to extend a Tutte path given for just a subgraph can be extended to a Tutte path of the entire graph. We then show how this technique can be utilized to prove Thomassen's Theorem 2.1 and also Sander's result (Theorem 2.2) such that only small overlaps occur.

Whenever we prove the existence of a Tutte path in a plane graph, we will do it by induction on the number of vertices in the given graph. The induction base will always be a triangle, for which the desired Tutte paths can be found trivially; thus, we will assume in these proofs by the induction hypothesis that graphs with fewer vertices contain Tutte paths. All graphs appearing in the inductive proof will be simple. Note that if we are given endvertices x, y and intermediate edge α such that $\alpha = xy$, the desired path is simply xy; thus, we assume $\alpha \neq xy$. Also, since G is 2-connected, C_G must be a cycle.

4.1 Two Easy Cases

For the cases covered in this section, let G be a simple plane 2-connected graph with outer face C_G and let $x \in V(G)$, $\alpha \in E(C_G)$ and $y \in V(G) - x$ be given as part of the input. We say that G is *decomposable into G_L and G_R* if it contains subgraphs G_L and G_R such that $G_L \cup G_R = G$, $V(G_L) \cap V(G_R) = \{c, d\}$, $x \in V(G_L)$, $\alpha \in E(G_R)$, $V(G_L) \neq \{x, c, d\}$ and $V(G_R) \neq \{c, d\}$ (or the analogous setting with y taking the role of x) (see Figure 4.2). In particular, $G_L \neq \{c, d\}$, even if $x \in \{c, d\}$. Hence, $\{c, d\}$ is a 2-separator of G. There might exist multiple pairs (G_L, G_R) into which G is decomposable; we will always choose a pair that minimizes $|V(G_R)|$. Note that G_R intersects C_G (for example, in α), but G_L does not have to intersect C_G. In [70], it was shown that any 2-connected plane graph G that is decomposable into G_L and G_R contains a Tutte path, without using recursion on overlapping subgraphs. It turns out the statement is independent of whether x and y are in C_G.

Lemma 4.1 ([70]). *If G is decomposable into G_L and G_R, then G contains an x-α-y-path.*

Proof. Let G'_L and G'_R be the plane graphs obtained from G_L and G_R, respectively, by adding the edge cd, if this does not already exist (see Figure 4.2). Let G^*_R be the graph obtained from G'_R by subdividing cd with a new vertex z. Clearly, all of the graphs G'_L, G'_R and G^*_R are 2-connected and contain fewer vertices than G.

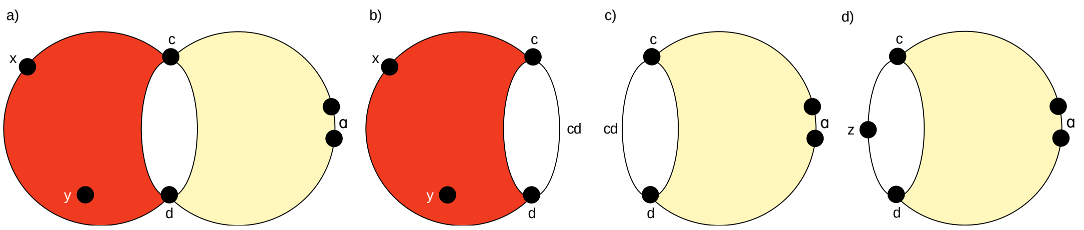


Figure 4.2: a) shows a graph G that is decomposable into G_L and G_R. The figures b) to d) show the graphs G'_L, G'_R and G^*_R (in this order).

Assume first that $y \in G_L$. By induction, G'_L contains an x-cd-y-path P_L and G'_R contains a c-α-d-path $P_R \not\ni cd$ (this requires to find a plane embedding of $G_{R'}$ whose outer face contains α; here and later, such an embedding can always be found by stereographic projection). Then $P := (P_L - cd) \cup P_R$ is an x-α-y-path of G, as $\{c, d\}$ is a 2-separator, and thus every P_L-bridge of G'_L and every P_R-bridge of G'_R has the same attachments as its corresponding P-bridge of G.

Otherwise, $y \in G_R - \{c, d\}$. We split this case in two sub-cases. First, assume $x \in \{c, d\}$ and without loss of generalization $x = c$. By induction, G'_R contains an x-α-y-path P_R. Suppose P_R does not contain d. Then d is contained in a P_R-bridge K of G'_R as internal vertex and $cd \in K$. Since $cd \in C_{G'_R}$, K has exactly two attachments (one of which is x), and these form a 2-separator implying that G is decomposable into a smaller graph than G_R, which contradicts our choice of the decomposition. Hence, $d \in P_R$. If $cd \notin P_R$, P_R is a Tutte path of G, as $d \in P_R$ implies that $G_L - cd$ is a P_R-bridge of G having two attachments. If $cd \in P_R$, let e be any edge in $G_L \cap C_G$; by induction, G_L contains a c-e-d-path P_L. Then $P_L \cup (P_R - cd)$ is an x-α-y-path of G.

Now assume $x \notin \{c, d\}$. We will again merge two Tutte paths by induction, but have to ensure that cd is not contained in any of them; to this end, we use G^*_R instead of G'_R. By induction, there is a z-α-y-path P_R in G^*_R; P_R contains either zc or zd, say without loss of generalization zc. By the same argument as in the previous case, we have $d \in P_R$. By induction, G'_L contains an x-cd-d-path P_L. Then $P := (P_L - d) \cup (P_R - z)$ is an x-α-y-path of G, as $\{c, d\} = P_L \cap P_R$ and since every P_L- or P_R-bridge of G_L or G_R, respectively, has the same attachments as its corresponding P-bridge of G. □

Even if G is not decomposable into G_L and G_R, G may contain other 2-separators $\{c, d\}$ that allow for a similar reduction as in Lemma 4.1 (for example, when modifying its prerequisites to satisfy $\{x, \alpha, y\} \subseteq G_R - \{c, d\}$). We give our own proof as the following lemma is not explicitly stated in [70].

Lemma 4.2. *Let $\{c, d\}$ be a 2-separator of G and let J be a $\{c, d\}$-bridge of G having an internal vertex in C_G such that x, y and α are not in J. Then G contains an x-α-y-path.*

Proof. Let G' be the plane graph obtained from G by deleting all internal vertices of J. Since $x \notin J$, G' contains at least three vertices. First, consider the case $E(C_G) - E(J) = \{\alpha\}$. Then G' is 2-connected, as the 2-connectivity of G and the deletion of the internal vertices of J for G' imply that any 1-separator z of G' must separate c from d. By induction, G' contains an x-α-y-path P. Since $c, d \in P$ and J has two attachments, P is also an x-α-y-path of G.

In the remaining case $E(C_G) - E(J) \neq \{\alpha\}$, we add the edge cd to G' where $C_G \cap J$ used to be embedded, unless cd is already contained in G'. Clearly, G' is 2-connected and $|V(G')| < n$, since J contains an internal vertex. By induction, G' contains an x-α-y-path P. If $cd \notin P$, cd is contained in a P-bridge of G' that has two attachments and its corresponding P-bridge of G has exactly the same attachments, so that P is also an x-α-y-path of G.

Now assume $cd \in P$ and let $J^* := J \cup \{cd\}$ such that cd is embedded where $G - V(J)$ used to be embedded. Then J^* is 2-connected and $|V(J^*)| < n$. Let α_{J^*} denote an arbitrary edge in $C_{J^*} - cd$. By induction, J^* contains a c-α_{J^*}-d-path P_{J^*}. Then the path obtained from P by replacing cd with P_{J^*} is an x-α-y-path of G, as $\{c, d\}$ separates the P- and P_{J^*}-bridges of G. □

For simplicity, we will call a graph *non-decomposable* if we can neither apply Lemma 4.1 nor Lemma 4.2 to it.

4.2 Moving from a Chain of Blocks to the Entire Graph

In this section, we will assume that we are given a path $Q := q_1C_Gq_2$ with endvertices q_1 and q_2 and a Tute path P in a plane chain of blocks in $G - Q$. We will then show how to modify Q such that any $(P \cup Q)$-bridge of G has at most three attachments and two if it contains an edge of $q_1C_Gq_2$. As $P \cup Q$ is not necessarily connected, this modification will not immediately result in a Tutte path of G, but as it was shown in the original proofs for Theorem 2.1 and Theorem 2.2, if we choose the endvertices of P and Q depending on x, y and α it is easy to connect P and Q such that their union is a Tutte path of G. The details of this will be covered in the following sections.

To be more formal, let G be a 2-connected plane graph. Let $p_1 \neq p_2$ be two vertices in $V(G) \setminus \{q_1, q_2\}$. Let K be a plane chain of blocks in $G - Q$ that contains p_1 and p_2 and let P be a Tutte path of K from p_1 to p_2. In addition, let K and Q be such that $V(Q \cup (C_G \cap K)) = V(C_G)$. Let $T := P \cup Q$, we will next show how to modify T. Consider any nontrivial T-bridge J of G. Let $C_G(J)$ denote the shortest path in $C_G \cap Q$ that contains all vertices in $J \cap Q$. Let l_J be the endvertex of $C_G(J)$ closest to q_1 and let r_J be the other endvertex of $C_G(J)$ If J has all of its attachments in $C_G(J)$, then $|V(J \cup C_G(J))| < |V(G)|$, $J \cup C_G(J)$ is 2-connected and by Theorem 2.2 contains a $l_J - r_J$ path Q_J. In this case we modify T by replacing l_JQr_J with Q_J. Note that this modification does not change the number of attachments for any Q_J-bridge of G nor any other T-bridge of G as the neighborhood of any vertex in $J \cup C_G(J) - \{l_J, r_J\}$ is the same as in G. On the other hand, if J has all of its attachments in $P \subseteq K$ it follows that $J \subseteq K$.

Lemma 4.3. *If J has no attachments in Q, then $J \subseteq K$ and J has at most three attachments in P.*

Proof. Let P_J denote the shortest connected path in P that contains all attachments of J. Note that $J \cup P_J$ must be 2-connected as any of its 1-separators would also be a 1-separator of G (contradicting that G is 2-connected). In addition, J's attachments are all in $P \subseteq K$ and the blocks in K are maximal in $G - Q$. Therefore, it follows that J must be a subgraph of K. As P is a Tutte path of K, J must have at least two and at most three attachments in P. □

Hence, by Lemma 4.3, to show that any T-bridge of G has at most three attachments and exactly two if it contains an edge of $q_1C_Gq_2$, it suffices to only consider T-bridges that have attachments in both P and Q. The following lemma showcases some properties of these T-bridges of G (also see Figure 4.3 for an illustration).

Lemma 4.4. *Let G be a 2-connected plane graph, Q a path in C_G, K a plane chain of blocks of $G - Q$ and P a Tutte subgraph of K. If J is a $(P \cup Q)$-bridge of G that has at least one attachment in both P and Q, then either $J \cap K$ is one vertex in P or J contains exactly one nontrivial outer P-bridge L of K. In particular, J has at most two attachments in P.*

Proof. If J does not contain an internal vertex of any P-bridge of K, then J can have at most one attachment in P. Assume for contradiction that J does have another attachment in P. By the definition of bridges, there must exist a path in J (and therefore in $G - Q$)

connecting both attachments. As we assume that J does not intersect K, other than in its attachments in P, this path contradicts the maximality of the blocks in K. Therefore, in the case where J does not contain an internal vertex of any P-bridge of K, $J \cap K$ is exactly the attachment of J in P.

Next, we assume that J contains at least one internal vertex v of some P-bridge L of K. We first prove that there is no other P-bridge of K contained in J. Assume to the contrary that there exists a vertex v' in J that is also part of a second P-bridge $L' \neq L$ of K. By definition the internal vertices of J induce a connected subgraph of G, therefore $J - (P \cup Q)$ contains a path from v to v'. As this path is also a subgraph of $G - Q$ and it connects two vertices in K, itself must also be part of K. The existence of such a path contradicts that L and L' are actually two different P-bridges of K.

It remains to show that L is an outer P-bridge of K and therefore has exactly two attachments in P. This follows from the assumption that the embedding of K was not changed from the embedding of G when computing P and the fact that G is planar. As v is part of L and $L \subseteq K$ any path from v to a vertex in C_G must intersect C_K in at least one vertex. If for all such paths all intersections with C_K are also in P, then $J - (P \cup Q)$ would not be connected, and therefore contradict that a bridge like J even exits. Thus, there must exist a path from v to Q in G, which does not intersect P. As this path has to intersect C_K in some vertex, this vertex must be part of L in K as well. Therefore, L must be an outer P-bridge of K. □

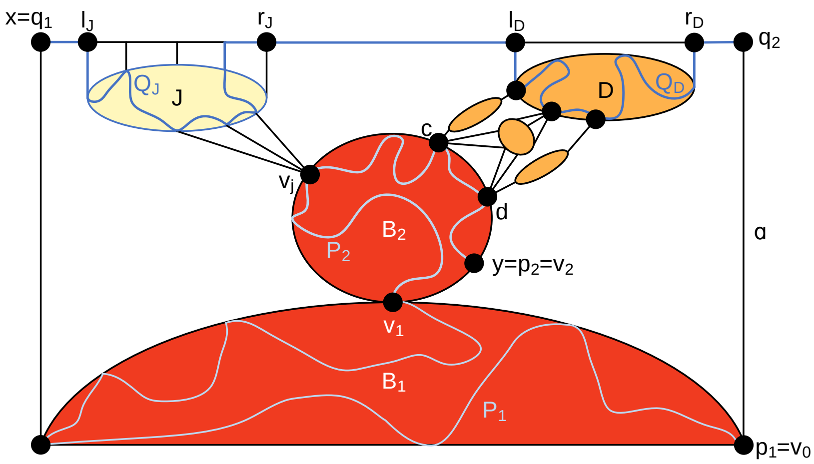


Figure 4.3: K consists of all subgraphs colored gray (B_1, B_2 and D). Here we have two $(P \cup Q)$-bridges J and D of G, J has exactly one attachment v_J in P, and D has exactly two attachments $\{c, d\}$ in P that are the attachments of a nontrivial outer P-bridge of K.

Because Lemma 4.2 is not applicable to G, there is no other T-bridge than J that intersects $(J \cup C_G(J)) - P - \{l_J, r_J\}$; in other words, $J \cup C_G(J)$ is everything that is enclosed by the attachments of J in G. In order to obtain the path T, we will compute a Tutte path Q_J of J from l_J to r_J such that any $(Q_J \cup P)$-bridge of G that intersects $(J \cup C_G(J)) - P - \{l_J, r_J\}$ has at most three attachments and at most two if it contains an edge of Q.

If $C_G(J)$ is a single vertex, we can set $Q_J := C_G(J)$, as then $J \cup C_G(J)$ does not contain an edge of Q and has at most three attachments in total (one in Q and at most two in P by Lemma 4.4). If $C_G(J)$ is not a single vertex, then by Lemma 4.4, it suffices to distinguish two cases, namely whether J has one or two attachments in P. In [70, 17], the following lemma was proven for fixed p_1, p_2, q_1 and q_2 in order to prove Theorem 2.1. We cover these cases in a more general form in order to be able to reuse the lemma in later proofs of this thesis.

Lemma 4.5. *Let G be a 2-connected plane graph, Q a connected subgraph of C_G and P a subgraph of $G-(V(Q)\setminus\{q_1,q_2\})$ and J be any $(P\cup Q)$-bridge of G that has either one or two attachments in P and at least two in Q. Then $(J\cup C_G(J))-P$ contains a path Q_J from l_J to r_J such that any $(Q_J\cup P)$-bridge of G that intersects $(J\cup C_G(J))-P-\{l_J,r_J\}$ has at most three attachments and at most two if it contains an edge of C_G.*

Proof. Assume first that J has only one attachment v in P (see Figure 4.3). Let $J' := J \cup C_G(J) \cup \{r_J v\}$ (without introducing multi-edges). Note that $|V(J')| < |V(G)|$ and that J' is 2-connected. The first claim simply follows from the fact that $|V(P)| \geq 2$ and $J*$ intersects P in only one vertex.

For proving that J' is 2-connected, consider the outer face $C_{J'}$ of J' and let F be the unique inner face of G that contains v and l_J. Since G is 2-connected, F is a cycle, and hence $vC_{J'}l_J$ is a simple path in G. Therefore, $C_{J'}$ is actually the union of $vC_{J'}l_J$, $C_G(J)$ and $\{r_J v\}$, which implies that $C_{J'}$ is a cycle. Hence, if we assume for contradiction that there exists a 1-separator w of J', then w must be in $J' - C_{J'}$. This assumption would also imply that there exists a component S in $J' - w$ that does not intersect the cycle $C_{J'}$. As J' and J differ at most by the edge $r_J v$, the neighborhood of S in G is would be the same as in J', which implies that w would also be a 1-separator in G. As this would contradict our assumption that G is 2-connected, J' must also be 2-connected.

By Theorem 2.1, J' contains a l_J-$r_J v$-v-path $Q_{J'}$. We set $Q_J := Q_{J'} - v$; then $Q_J \cap P = \emptyset$ and the neighborhood of every internal vertex of every $Q_{J'}$-bridge of J' is the same in J' as in G. Thus, every Q_J-bridge of G corresponds to a $Q_{J'}$-bridge of J', which ensures that the number of attachments of every Q_J-bridge of G intersecting $(J \cup C_G(J)) - P - \{l_J, r_J\}$ is as claimed.

Assume now that J has exactly two attachments c and d in P. Since J is connected and contains no edge of $C_G(J)$, there must exist some cycle in $J \cup C_G(J)$ that contains $C_G(J)$. Since G is 2-connected and this cycle is also part of G, the subgraph of G induced by the vertices in this cycle and the vertices of G embedded inside this cycle must be 2-connected as well. This implies that there exists a block D in $J \cup C_G(J)$ that contains all vertices of $C_G(J)$(see Figure 4.3).

Consider a $(D \cup \{c,d\})$-bridge L' of $J \cup C_G(J)$. Then L' has at least one attachment in D, as otherwise L' itself would be a $\{c,d\}$-bridge of G, which contradicts that L' is contained in $J \cup C_G(J)$. Moreover, L' has exactly one attachment in D, as a second attachment would contradict the maximality of D. By planarity, there is at most one $(D \cup \{c,d\})$-bridge L that has three attachments c, d and, say, $v_L \in D$.

We distinguish two cases. If L exists, set $v_D := v_L$. If L does not exist, let R be the minimal path in $C_D - inner(C_G(J))$ that contains the attachments of all $(D \cup \{c,d\})$-bridges of J that are in D. Then R contains a vertex v_D that splits R into two paths R_c and R_d such that $R_c \cap R_d = \{v_D\}$. Moreover, any $(D \cup \{c,d\})$-bridge of J having c as one of its two attachments has its other attachment in R_c, and any $(D \cup \{c, d\})$-bridge of J having d as one of its two attachments has its other attachment in R_d. In either case for the vertex v_D, we define β as an edge of C_D that is incident to v_D.

As D is 2-connected, by Theorem 2.1 there exists an l_J-β-r_J-path Q_D of D. Any outer Q_D-bridge of D therefore maybe gain either c or d as third attachment when considering this bridge in G, but not both; if L exists, L has still only the three attachments $\{c, d, v_L\}$ in G. Thus, Q_D is the desired path Q_J. □

We replace $l_J C_G r_J$ in T with Q_J for every $(P \cup Q)$-bridge J. Since l_J and r_J are contained in T, no $(P \cup Q)$-bridge of G other than J is affected by this "local" replacement, which proves its sufficiency for obtaining the desired path Q.

4.3 A Constructive Proof for Thomassen's Result

We now prove that any 2-connected plane graph G contains an $x - \alpha - y$ path for any $x \in V(C_G), y \in V(G) - x$ and $\alpha \in E(C_G)$. For simplicity, if y is not in $V(C_G)$ but has degree two and both of its neighbors are in $V(C_G)$, then we change the embedding of G (, and therefore C_G) such that y belongs to the outer face. If Lemma 4.1 or Lemma 4.2 can be applied, we obtain such a Tutte path directly, so assume their prerequisites are not met. Let l_α be the endvertex of α that appears first when we traverse C_G in clockwise order starting from x, and let r_α be the other endvertex of α. If $y \in xC_G l_\alpha$, we interchange x and y (this does not change l_α); hence, we have $y \notin xC_G l_\alpha$. If $y = r_\alpha$, we mirror the embedding such that y becomes l_α and proceed as in the previous case; hence, $y \notin xC_G r_\alpha$.

In order to apply the technique showcased in Section 4.2, we define two paths P and Q in G, whose union will be modified into a Tutte path of G. Let $Q := xC_G l_\alpha$ and let $H := G - V(Q)$; in particular, $y \notin Q$ and, if x is an endvertex of α, $Q = \{x\}$. Since G is non-decomposable, we have $deg(r_\alpha) \geq 3$, as otherwise the neighborhood of r_α would be the 2-separator of such a decomposition. Since $deg(r_\alpha) \geq 3$, r_α is incident to some edge $e \notin C_G$ that shares a face with α. Let B_1 be the block of H that contains e. It is straight-forward to prove the following about B_1 (see Thomassen [70]), which shows that every vertex of C_G is either in Q or in B_1.

Lemma 4.6 ([70])**.** *B_1 contains $C_G - V(Q)$ and is the only block of H containing r_α.*

Consider a component A of H that does not intersect B_1. Then all vertices in the neighborhood of A in G must be in Q. This implies that there exists a subpath in Q that contains all neighbors of A in G and its endvertices form a 2-separator of G. Hence, either $y \in A$ and we can apply Lemma 4.1 or $y \notin A$ and we can apply Lemma 4.2. Since both contradicts our assumptions, H is connected and contains B_1 and y. Let K be the minimal plane chain of blocks $B_1, \ldots, B_l$ of H that contains B_1 and y (hence, $y \in B_l$). Let v_i be the intersection of B_i and B_{i+1} for $1 \leq i \leq l - 1$; in addition, we set $v_0 := r_\alpha$ and $v_l := y$.

Consider any $(K \cup C_G)$-bridge J. Since Lemma 4.2 cannot be applied to G, J has an attachment $v_J \in K$. Further, J cannot have two attachments in K, as this would contradict the maximality of the blocks in K.

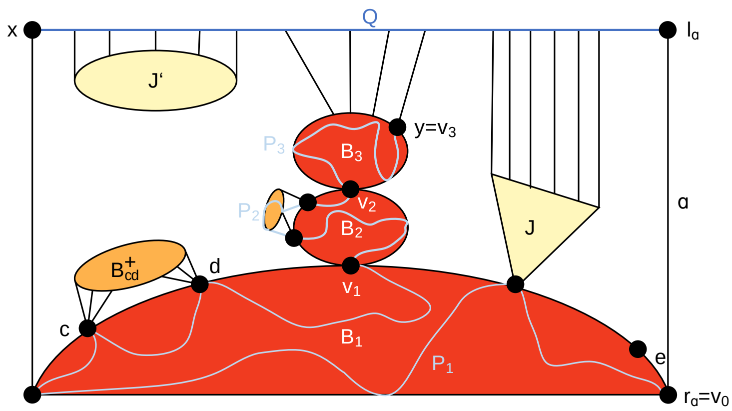


Figure 4.4: The paths Q and $P = P_1 \cup P_2 \cup P_3$, the subgraph H of G and its minimal chain of blocks $K = B_1 \cup B_2 \cup B_3$, and a $(K \cup C_G)$-bridge J. A $(K \cup C_G)$-bridge like J' cannot exist due to Lemmas 4.1 and 4.2.

4.3.1 Decomposing along Maximal 2-Separators

At this point we will deviate from the original proof of Theorem 2.1 in [70], which continues with induction on every block of K that leads to overlapping subgraphs in a later step of the proof. Instead, we will show that a v_0-v_l-path P of K can be found iteratively such that the graphs in the induction have only small overlap.

For every block $B_i \neq B_1$ of K, we choose an arbitrary edge $\alpha_i = l_{\alpha_i} r_{\alpha_i}$ in C_{B_i}. In B_1 we choose α_1 such that α_1 is incident to the endvertex of $C_{B_1} \cap C_G$ that is not r_α. As done for G, we may assume for every B_i that l_{α_i} is the endvertex of α_i that is contained in $v_{i-1} C_{B_i} \alpha_i$ and that $v_i \notin v_{i-1} C_{B_i} r_{\alpha_i}$ and (by mirroring the planar embedding and interchanging v_i and v_{i-1} if necessary). However, unlike G, some B_i may satisfy the prerequisites of Lemmas 4.1 and 4.2. Note that by the induction hypothesis of Theorem 2.1, B_i contains a v_{i-1}-α_i-v_i-path P_i, but we do not apply the induction hypothesis just yet. In [70] applying the induction hypothesis at this point results in the fact that the outer P_i-bridges of B_i are not only being processed here but also in a later induction step when modifying Q. We avoid such overlapping subgraphs by using a new iterative structural decomposition of B_i along certain 2-separators on C_{B_i}. This decomposition allows us to construct P_i iteratively such that the outer P_i-bridges of B_i are not part of the induction applied on B_i. Eventually, $P := \bigcup_{1 \leq i \leq l} P_i$ will be the desired v_0-v_l-path of K.

The outline is as follows. After explaining the basic split operation that is used by our decomposition, we give new insights into the structure of the Tutte paths P_i of the blocks B_i. These are used in Section 4.3.2 to define the iterative decomposition of every block B_i into a modified block $\eta(B_i)$, which will in turn allow to compute every P_i step-by-step. This gives the first part P of the desired x-α-y path of G. Subsequently, we will use Lemma 4.5, as outlined in Section 4.2, to obtain the second part.

For a 2-separator $\{c, d\} \subseteq C_B$ of a block B, let B^+_{cd} be the $\{c, d\}$-bridge of B that contains $cC_B d$ and let B^-_{cd} be the union of all other $\{c, d\}$-bridges of B (note that B^+_{cd} contains the edge cd if and only if B^+_{cd} is trivial); see Figure 4.4. For a 2-separator $\{c, d\} \subseteq C_B$, let *splitting off* B^+_{cd} (from B) be the operation that deletes all internal vertices of B^+_{cd} from B and adds the edge cd if cd does not already exist in B. Our decomposition proceeds by iteratively splitting off bridges B^+_{cd} from the blocks B_i of K for suitable 2-separators $\{c, d\} \subseteq C_{B_i}$ (we omit the subscript i in such bridges B^+_{cd}, as it is determined by c and d). The following lemma restricts these 2-separators to be contained in specific parts of the outer face.

Lemma 4.7. *Let P' be a Tutte path of a block B. For any two vertices a and b in $P' \cap C_B$, any outer P'-bridge J of B has both attachments in aC_Bb or both in bC_Ba. If additionally J is nontrivial and $P' \neq ab$, the attachments of J form a 2-separator of B.*

Proof. Note that the first claim is trivially true if at least one of J's attachments is in $\{a, b\}$, therefore we assume that J has attachments $c, d \notin \{a, b\}$. As J is an outer P'-bridge of B, we know that c and d are both in C_B. Further, as C_B is a cycle and removing two vertices from a cycle can produce at most two components, we know that a and b must be in the same component of $C_B - \{c, d\}$. Therefore, this component contains either aC_Bb or bC_Ba, and thus c, d must be in either aC_Bb or bC_Ba, respectively. For the second claim, let z be an internal vertex of J. Since $P' \neq ab$, P' contains a third vertex $v \notin \{a, b\}$. As v is not contained in J, $\{c, d\}$ separates z and v and is thus a 2-separator of B. □

For every block $B_i \neq B_l$ of K, let the *boundary points* of B_i be the vertices v_{i-1},l_{α_i},r_{α_i} and v_i, and let the *boundary parts* of B_i be the inclusion-wise maximal paths of C_{B_i} that do not contain any boundary point as inner vertex (see Figure 4.5a; note that boundary parts may be single vertices). Hence, every boundary point will be contained in any possible v_{i-1}-α_i-v_i-path P_i, and there are exactly four boundary parts, one of which is α_i. Now, if $P_i \neq \alpha_i$, applying Lemma 4.7 for all boundary points $a, b \in \{v_{i-1}, l_{\alpha_i}, r_{\alpha_i}, v_i\}$ and $\alpha' := \alpha_i$ implies that the two attachments of every outer nontrivial P_i-bridge of B_i form a 2-separator that is contained in one boundary part of B_i. For this reason, our decomposition will split off only 2-separators that are contained in boundary parts.

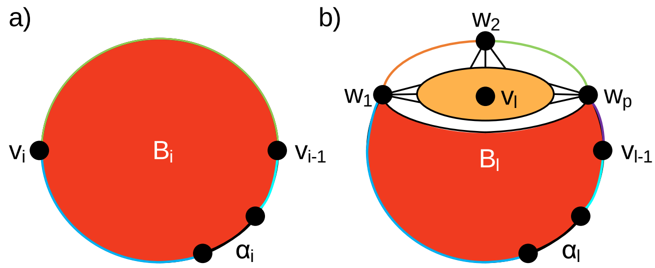


Figure 4.5: a) The boundary points and parts of a block $B_i \neq B_l$. b) An instance in which the block B_l contains a 2-separator $\{w_1, w_p\}$ that splits off v_l.

In principle, we will do the same for the block B_l. If $v_l \in C_{B_l}$, we define the boundary points of B_l just as before for $i < l$. However, B_l is special in the sense that v_l may not be in C_{B_l}. Then we have to ensure that we do not loose v_l when splitting off a 2-separator, as v_l is supposed to be contained in P_l (see Figure 4.5b). To this end, consider for $v_l \notin C_{B_l}$ the 2-separator $\{w_1, w_p\} \subseteq C_{B_l}$ of B_l such that $B^+_{w_1,w_p}$ contains v_l, the path $w_1C_{B_l}w_p$ is contained in one of the paths in $\{v_{l-1}C_{B_l}\alpha_l, \alpha_l, \alpha_lC_{B_l}v_{l-1}\}$ and $w_1C_{B_l}w_p$ is of minimal length if such a 2-separator exists. The restriction to these three parts of the boundary is again motivated by Lemma 4.7: If $P_l \neq \alpha_l$ and there is an outer nontrivial P_l-bridge of B_l, its two attachments are in P_l, and thus we only have to split off 2-separators that are in one of these three paths to avoid these P_l-bridges in the induction. If the 2-separator $\{w_1, w_p\}$ exists, let $w_1, \ldots, w_p$ be the $p \geq 2$ attachments of the $w_1 C_{B_l} w_p$-bridge of B_l that contains v_l, in the order of appearance in $w_1 C_{B_i} w_p$; otherwise, let for notational convenience $w_1 := \cdots := w_p := l_{\alpha_i}$. In the case $v_l \notin C_{B_l}$, let the *boundary points* of B_l be $v_{l-1}, l_{\alpha_l}, r_{\alpha_l}, w_1, \ldots, w_p$ and let the *boundary parts* of B_l be the inclusion-wise maximal paths of C_{B_l} that do not contain any boundary point as inner vertex.

Lemma 4.8. *If the 2-separator $\{w_1, w_p\}$ exists, it is unique and every v_{l-1}-α_l-v_l-path P_l of B_l contains the vertices $w_1, \ldots, w_p$.*

Proof. Let $J \subset B^+_{w_1, w_p}$ be the $w_1 C_{B_l} w_p$-bridge of B_l that contains v_l and has attachments $w_1, \ldots, w_p$. For the first claim, assume to the contrary that there is a 2-separator $\{w'_1, w'_{p'}\} \neq \{w_1, w_p\}$ of B_l having the same properties as $\{w_1, w_p\}$. By the connectivity of J and the property that restricts $\{w'_1, w'_{p'}\}$ to the three parts of the boundary of B_l, $\{w'_1, w'_{p'}\}$ may only split off a subgraph containing v_l if $w_1 C_{B_l} w_p \subset w'_1 C_{B_l} w'_{p'}$. This however contradicts the minimality of the length of $w'_1 C_{B_l} w'_{p'}$.

For the second claim, let P_l be any v_{l-1}-α_l-v_l-path of B_l. Assume to the contrary that $w_j \notin P_l$ for some $j \in \{1, \ldots, p\}$. Then w_j is an internal vertex of an outer P_l-bridge J' of B_l. By Lemma 4.7, both attachments of J' are in C_{B_l}. However, since J contains a path from $w_j \notin P_l$ to $v_l \in P_l$ in which only w_j is in C_{B_l}, at least one attachment of J' is not in C_{B_l}, which gives a contradiction. □

Lemma 4.8 ensures that the boundary points of any B_i are contained in every Tutte path P_i of B_i. Every block $B_i \neq B_l$ has exactly four boundary parts and B_l has at least three boundary parts (three if $v_l \notin C_{B_l}$ and $\{w_1, w_p\}$ does not exist), some of which may have length zero. For every $1 \leq i \leq l$, the boundary parts of B_i partition C_{B_i}, and one of them consists of α_i. This implies in particular that B_i has at least two boundary parts of length at least one unless $B_i = \alpha_i$. We need some notation to break symmetries on boundary parts. For a boundary part Z of a block B, let $\{c, d\}^* \subseteq Z$ denote two elements c and d (vertices or edges) such that cC_Bd is contained in Z (this notation orders c and d consistently to the clockwise orientation of C_B); if cC_Bd is contained in some boundary part of B that is not specified, we just write $\{c, d\}^* \subseteq C_B$.

We now define which 2-separators are split off in our decomposition. Let a 2-separator $\{c, d\}^* \subseteq C_B$ of B be *maximal in a boundary part Z of B* if $\{c, d\} \subseteq Z$ and Z does not contain a 2-separator $\{c', d'\}$ of B such that $cC_Bd \subset c'C_Bd'$. Let a 2-separator $\{c, d\}^* \subseteq C_B$ of B be *maximal* if $\{c, d\}^*$ is maximal with respect to at least one boundary part of B. Hence, every maximal 2-separator is contained in a boundary part, and 2-separators that are contained in a boundary part are maximal if they are not properly "enclosed" by other 2-separators on the same boundary part.

Let two maximal 2-separators $\{c, d\}^*$ and $\{c', d'\}^*$ of B *interlace* if $\{c, d\} \cap \{c', d'\} = \emptyset$ and their vertices appear in the order c, c', d, d' or c', c, d', d on C_B (in particular, both 2-separators are contained in the same boundary part of B). In general, maximal 2-separators of a block B_i of K may interlace; for example, consider the two maximal 2-separators when B_i is a cycle on four vertices in which v_{i-1} and v_i are adjacent.

However, the following lemma shows that such interlacing is only possible for very specific configurations.

Lemma 4.9. *Let $\{c,d\}^*$ and $\{c',d'\}^*$ be interlacing 2-separators of B_i in a boundary part Z such that $c' \in cC_{B_i}d$ and at least one of them is maximal. Then $d'C_{B_i}c = v_{i-1}v_i = \alpha_i$.*

Proof. Since $\{c,d\}$ is a 2-separator, $B_i - \{c,d\}$ has at least two components. We argue that there are exactly two. Otherwise, $B_i - \{c,d\}$ has a component that contains the inner vertices of a path P' from c to d in $B_i - (C_{B_i} - \{c,d\})$. Then $B_i - \{c',d'\}$ has a component containing $(P' \cup C_{B_i}) - \{c',d'\}$ and no second component, as this would contain the inner vertices of a path from c' to d' in $B_i - ((P' \cup C_{B_i}) - \{c',d'\})$, which does not exist due to planarity. Since this contradicts that $\{c',d'\}$ is a 2-separator, we conclude that $B_i - \{c,d\}$, and by symmetry $B_i - \{c',d'\}$, have exactly two components.

By the same argument, $inner(cC_{B_i}d)$ and $inner(dC_{B_i}c)$ are contained in different components of $B_i - \{c,d\}$ and the same holds for $inner(c'C_{B_i}d')$ and $inner(d'C_{B_i}c')$ in $B_i - \{c',d'\}$. Hence, the component of $B_i - \{c,d'\}$ that contains $inner(cC_{B_i}d') \neq \emptyset$ does not intersect $inner(d'C_{B_i}c)$. If $inner(d'C_{B_i}c) \neq \emptyset$, this implies that $\{c,d'\} \subseteq Z$ is a 2-separator of B_i, which contradicts the maximality of $\{c,d\}$ or of $\{c',d'\}$. Hence, $inner(d'C_{B_i}c) = \emptyset$, which implies that $d'C_{B_i}c$ is an edge. As Z is not an edge, $d'C_{B_i}c = \alpha_i$. Since c and d' are the only boundary points of B_i, either $\{c,d'\} = \{v_{i-1},v_i\}$ or $B_i = B_l$, $v_l \notin C_{B_l}$, $\{c,d'\} = \{v_{i-1},w_2\}$, $v_{i-1} = w_1$ and $w_2 = w_p$. However, the latter case is impossible, as then $\{c,d'\}$ would be a 2-separator that separates $inner(cC_{B_i}d') \neq \emptyset$ and v_l, which contradicts the maximality of $\{c,d\}$ or of $\{c',d'\}$. This gives the claim. □

If two maximal 2-separators interlace, Lemma 4.9 thus ensures that these two are the only maximal 2-separators that may contain v_{i-1} and v_i, respectively. This gives the following direct corollary.

Corollary 4.10. *Every block of K has at most two maximal 2-separators that interlace.*

Note that any boundary part may nevertheless contain arbitrarily many (pairwise non-interlacing) maximal 2-separators. The next lemma strengthens Lemma 4.7.

Lemma 4.11. *Let P_i be a v_{i-1}-α_i-v_i-path of B_i. Let J be a nontrivial outer P_i-bridge of B_i and let e be an edge in $J \cap C_{B_i}$. Then the attachments of J are contained in the boundary part of B_i that contains e.*

Proof. Let c and d be the attachments of J such that $e \in cC_{B_i}d$ and let Z be the boundary part of B_i that contains e. If $P_i = \alpha_i$, $v_{i-1} = l_{\alpha_i}$ and $v_i = r_{\alpha_i}$ are the only boundary points of B_i. Then c and d are the endvertices of $Z = v_iC_{B_i}v_{i-1} \ni e$, which gives the claim.

Otherwise, let $P_i \neq \alpha_i$. By applying Lemma 4.7 with $a = l_{\alpha_i}$ and $b = r_{\alpha_i}$, $\{c,d\}$ is a 2-separator of B_i that is contained in C_{B_i}. By definition of $w_1,\dots,w_p$, there are at least three independent paths between every two of these vertices in B_i; thus, $\{c,d\}$ does not separate two vertices of $\{w_1,\dots,w_p\}$. Since all other possible boundary points $(v_{i-1}, l_{\alpha_i}, r_{\alpha_i}, v_i)$ are contained in P_i, applying Lemma 4.7 on these implies that $\{c,d\}$ does not separate two vertices of these remaining boundary points. Hence, if $\{c,d\} \not\subseteq Z$, we have $B_i = B_l$ and $v_l \notin C_{B_l}$ such that $\{c,d\}$ separates $\{w_1,\dots,w_p\}$ from the remaining boundary points. Since the P_i-bridge J does not contain $\alpha_l \in P_i$, $cC_{B_l}d \subseteq J$ contains

$\{w_1, \ldots, w_p\}$, but $inner(cC_{B_l}d)$ does not contain any other boundary point. As $v_l \in P_i$, at least one of $\{w_1, w_p\}$ must be in P_i, say w_p by symmetry. Then $d = w_p$, as $w_p \in P_i$ cannot be an internal vertex of J. Now, in both cases $p = 2$ (which implies $c \neq w_1$, as $\{c, d\} \nsubseteq Z = w_1 C_{B_l} w_2$) and $p \geq 3$, J contains the edge of P_i that is incident to v_l. As this contradicts that J is a P_i-bridge, we conclude $\{c, d\} \subseteq Z$. □

Now we relate nontrivial outer P_i-bridges of B_i to maximal 2-separators of B_i. In the next subsection, we will use this lemma as a fundamental tool for a decomposition into subgraphs having only small overlaps, which will eventually construct P.

Lemma 4.12. *Let P_i be a v_{i-1}-α_i-v_i-path of B_i such that $P_i \neq \alpha_i$. Then the maximal 2-separators of B_i are contained in P_i and do not interlace pairwise. If J is a nontrivial outer P_i-bridge of B_i, there is a maximal 2-separator $\{c, d\}^*$ of B_i such that $J \subseteq B^+_{cd}$.*

Proof. Consider the first claim. Since $P_i \neq \alpha_i$ implies $\alpha_i \neq v_{i-1}v_i$ by contraposition, no two maximal 2-separators interlace due to Lemma 4.9. Assume to the contrary that there is a maximal 2-separator $\{c, d\}^*$ of B_i such that c or d is not in P_i, say $c \notin P_i$ by symmetry (otherwise, we may flip B_i). Let Z be the boundary part of B_i that contains $\{c, d\}$. Now consider the nontrivial P_i-bridge J of B_i that contains c as internal vertex. Since $c \in Z$, J contains an edge of Z and is thus a nontrivial outer P_i-bridge. Let c' and d' be the attachments of J such that $c' C_{B_i} d' \subseteq J$. By Lemma 4.7, $\{c', d'\}$ is a 2-separator of B_i. By Lemma 4.11, $\{c', d'\} \subseteq Z$. Then Lemma 4.9 implies that $\{c', d'\}$ and the maximal 2-separator $\{c, d\}$ do not interlace. Since J contains the incident edge of c in $dC_{B_i}c$, we conclude $cC_{B_i}d \subset c' C_{B_i} d'$, which contradicts the maximality of $\{c, d\}$. This shows the first claim holds.

For the second claim, let c' and d' be the attachments of the given P_i-bridge J and let Z be the boundary part of B_i that contains some edge $e \in J \cap C_{B_i}$. By Lemma 4.7, $\{c', d'\}$ is a 2-separator of B_i. By Lemma 4.11, $\{c', d'\} \subseteq Z$. Hence, there is a maximal 2-separator $\{c, d\}^*$ of B_i in Z such that $\{c', d'\} \subseteq cC_{B_i}d$ and we conclude $J \subseteq B^+_{cd}$. □

4.3.2 The Construction of P

Naturally, we do not know the entire path P_i in B_i in advance. However, Lemma 4.12 ensures under the condition $P_i \neq \alpha_i$ that we can split off every nontrivial outer bridge J of P_i by a maximal 2-separator, no matter how P_i looks like. This allows us to construct P_i iteratively by decomposing B_i along its maximal 2-separators. Since maximal 2-separators only depend on the graph B_i , we can access them without knowing P_i itself. This fact also allows us to reuse this construction of P in other proofs, where we need to find a Tutte path of a given plane chain of blocks. We now show the details of such a decomposition given K.

Definition 4.13. *For every $1 \leq i \leq l$, let $\eta(B_i)$ be α_i if $\alpha_i = v_{i-1}v_i$ and otherwise the graph obtained from B_i as follows: For every maximal 2-separator $\{c, d\}^*$ of B_i, split off B^+_{cd}. Moreover, let $\eta(K) := \eta(B_1) \cup \cdots \cup \eta(B_l)$.*

If for all $B_i \in K$, $\alpha_i \neq v_{i-1}v_i$, then α_i cannot be a v_{i-1}-α_i-v_i-path of B_i; hence, the maximal 2-separators of K that were split in this definition do not interlace due to Lemma 4.12. This implies that the order in which the splits are performed is irrelevant.

In any case, we have $V(C_{\eta(B_i)}) \subseteq V(C_{B_i})$ and the only 2-separators of $\eta(B_i)$ must be contained in some boundary part of B_i, as there would have been another split otherwise. See Figure 4.6 for an illustration of $\eta(B_l)$. The following lemma highlights two important properties of every $\eta(B_i)$.

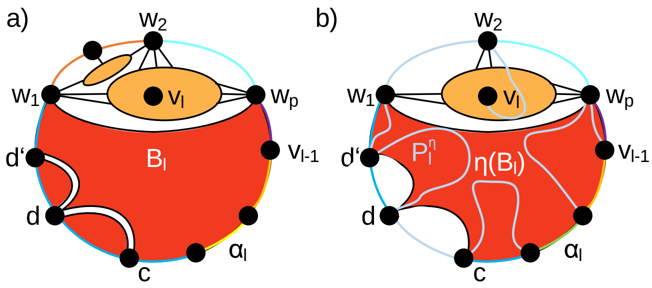


Figure 4.6: a) A block B_l with boundary points $v_{l-1}, l_{\alpha_l}, r_{\alpha_l}, w_1, \dots, w_3$ that has two maximal 2-separators on the same boundary part. b) The graph $\eta(B_l)$.

Lemma 4.14. *Every $\eta(B_i)$ is a block. Let P_i^η be a v_{i-1}-α_i-v_i-path of some $\eta(B_i)$ such that $P_i^\eta \neq \alpha_i$. Then every outer P_i^η-bridge of $\eta(B_i)$ is trivial.*

Proof. If $\alpha_i = v_{i-1}v_i$, $\eta(B_i) = \alpha_i$ is clearly a block. Otherwise, B_i has at least three vertices and is thus 2-connected; consider two independent paths in B_i between any two vertices in $\eta(B_i)$. Splitting off B_{cd}^+ for any maximal 2-separator $\{c, d\}^*$ (we may assume that not both independent paths are contained in B_{cd}^+) preserves the existence of such paths by replacing any subpath through B_{cd}^+ with the edge cd. Hence, $\eta(B_i)$ is a block.

For the second claim, we first prove that P_i^η contains all boundary points of B_i. By definition, P_i^η contains $l_{\alpha_i}, r_{\alpha_i}, v_{i-1}$ and v_i. The only possible remaining boundary points $w_1, \dots, w_p$ may occur only if $i = l$, $v_l \notin C_{B_l}$ and the 2-separator $\{w_1, w_p\}$ exists. In that case, we argue similarly as for Lemma 4.8: Let J be the $w_1C_{B_l}w_p$-bridge of B_l that contains v_l; clearly, J exists also in $\eta(B_l)$. Now assume to the contrary that $w_j \notin \eta(P_l)$ for some $j \in \{1, \dots, p\}$. Then w_j is an internal vertex of an outer $\eta(P_l)$-bridge J' of $\eta(B_l)$. As $\eta(B_l)$ is a block, we can apply Lemma 4.7, which implies that both attachments of J' are in $C_{\eta(B_l)}$. However, since J contains a path from $w_j \notin \eta(P_l)$ to $v_j \in \eta(P_l)$ in which only w_j is in $C_{\eta(B_l)}$, at least one attachment of J' is not in $C_{\eta(B_l)}$, which gives a contradiction.

Assume to the contrary that there is a nontrivial outer P_i^η-bridge J'' of $\eta(B_i)$ and let c, d be its two attachments. Lemma 4.7 implies that $\{c, d\}$ is a 2-separator of $\eta(B_i)$ that is contained in C_{B_i}. If c and d are contained in the same boundary part of B_i, a supergraph of B_{cd}^+ would therefore have been split off for $\eta(B_i)$, which contradicts that J'' is nontrivial. Hence, c and d are contained in different boundary parts of B_i. Then $inner(cC_{B_i}d)$ contains a boundary point of B_i and, as this boundary point is also in P_i^η, this contradicts that J'' is an outer P_i^η-bridge. □

The next lemma shows how we can construct a Tutte path P of K iteratively using maximal 2-separators. We will provide the details of an efficient implementation in Section 4.6.

Lemma 4.15 (Construction of P). *Given P_i^η for every $1 \leq i \leq l$, a v_{i-1}-α_i-v_i-path P_i of B_i can be constructed such that no nontrivial outer P_i-bridge of B_i is part of an inductive call of Theorem 2.1.*

Proof. The proof proceeds by induction on the number of vertices in B_i. If B_i is just an edge or a triangle, the claim follows directly. For the induction step, we therefore assume that B_i contains at least four vertices. If $\alpha_i = v_{i-1}v_i$, we set $P_i := \alpha_i$, so assume $\alpha_i \neq v_{i-1}v_i$. In particular, $\eta(B_i) \neq \alpha_i$ and α_i is no v_{i-1}-α_i-v_i-path of $\eta(B_i)$. As $|V(\eta(B_i))| < n$, we may apply an inductive call of Theorem 2.1 to $\eta(B_i)$, which returns a v_{i-1}-α_i-v_i-path $P_i^\eta \neq \alpha_i$ of $\eta(B_i)$. This does not violate the claim, since $\eta(B_i)$ does not contain any nontrivial outer P_i^η-bridge by Lemma 4.14.

Now we extend P_i^η iteratively to the desired v_{i-1}-α_i-v_i-path P_i of B_i by restoring the subgraphs that were split off along maximal 2-separators one by one. For every edge $cd \in C_{\eta(B_i)}$ such that $\{c,d\}^*$ is a maximal 2-separator of B_i (in arbitrary order), we distinguish the following two cases: If $cd \notin P_i^\eta$, we do not modify P_i^η, as in B_i the subgraph B_{cd}^+ will be a valid outer bridge. If otherwise $cd \in P_i^\eta$, we consider the subgraph B_{cd}^+ of B_i. Clearly, $B := B_{cd}^+ \cup \{cd\}$ is a block. Define that the *boundary points* of B are c, d and the two endvertices of some arbitrary edge $\alpha_B \neq cd$ in C_B. This introduces the boundary parts of B in the standard way, and hence defines $\eta(B)$. Note that B may contain several maximal 2-separators in cC_Bd that in B_i were suppressed by $\{c,d\}^*$, as $\{c,d\}^*$ is not a 2-separator of B. In consistency with Lemma 4.12, which ensures that no two maximal 2-separators of B_i interlace, we have to ensure that no two maximal 2-separators of B interlace in our case $\alpha_i \neq v_{i-1}v_i$, as otherwise $\eta(B)$ would be ill-defined. This is however implied by Lemma 4.9, as $\alpha_B \neq cd$. Since $|V(\eta(B))| < |V(B_i)|$, a c-α_B-d-path P_B of B can be constructed such that no nontrivial outer P_B-bridge of B is part of an inductive call of Theorem 2.1. Since $\alpha_B \neq cd$, P_B does not contain cd. We now replace the edge cd in P_i^η by P_B. This gives the desired path P_i after having restored all subgraphs B_{cd}^+. □

Applying Lemma 4.15 on all blocks of K and taking the union of the resulting paths gives the desired Tutte path P of K. Following the instructions in Section 4.2 we can modify Q such that $P \cup \{\alpha\} \cup Q$ becomes a Tutte path of G. By Lemma 4.15, no nontrivial outer P-bridge of K was part of any inductive call of Theorem 2.1 so far, which allows us to use these bridges inductively for the modification of Q via Lemma 4.5, while the existence proof in [70] used these arbitrarily large bridges in inductive calls for both constructing P and modifying Q.

4.4 The Three Edge Lemma

Next, we show how Thomassen's result implies the existence of a Tutte cycle through any three given edges in C_G and how we can use the tools developed in the previous sections to give a constructive proof. That such a Tutte cycle always exists was already proven in [67] or [58], and the result itself is known as the Three-Edge-Lemma. Our approach is novel in the fact that we can find this Tutte cycle without constructing overlapping subgraphs in the process.

Lemma 4.16 (Three-Edge-Lemma)**.** *Let G be a 2-connected plane graph and let α, β and γ be three arbitrary edges of C_G. There exists a Tutte cycle C in G that contains α, β and γ.*

Proof. We denote the endvertices of α, β and γ by $\{\alpha_1, \alpha_2\}, \{\beta_1, \beta_2\}$ and $\{\gamma_1, \gamma_2\}$ respectively such that $\alpha = \alpha_1 C_G \alpha_2, \beta = \beta_1 C_G \beta_2$ and $\gamma = \gamma_1 C_G \gamma_2$. Without loss of generality, we may also assume that α, β and γ appear in this order when traversing C_G starting from α_1 in clockwise direction (see Figure 4.7). The proof proceeds by induction on the number of vertices in G. In the base-case, G is a triangle and the claim is true.

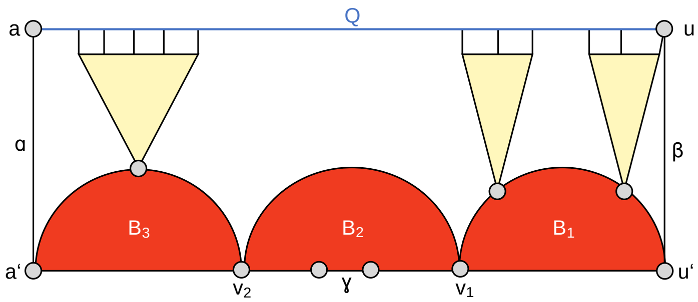


Figure 4.7: A graph G with edges α, β, γ that contains a plane chain of blocks K, as used in the Three Edge Lemma.

Let $Q := \alpha_2 C_G \beta_1$ and K be a minimal plane chain of blocks $B_1, \ldots, B_l$ of $G - Q$ that contains α_1 and β_2, and let $1 \leq k \leq l$ be the index such that $\gamma \in B_k$. Let $v_i := B_i \cap B_{i+1}$ for $2 \leq i \leq l-1$, $v_0 := \beta_2$ and $v_l := \alpha_1$. For every block $B_i \neq B_k$ in K let δ_i be an arbitrary edge in $B_i \cap C_G$. In B_k let $\delta_k := \gamma$. In addition, we denote the endvertices of δ_i by l_{delta_i} and r_{δ_i}. For every block B_i of K, let the boundary points of B_i be the vertices $v_{i-1}, l_{\delta_i}, r_{\delta_i}, v_i$ and let the boundary parts of B_i be the inclusion-wise maximal paths of C_{B_i} that do not contain any boundary point as inner vertex. Note that this suffices to define $\eta(B_i)$ for every i, which allows us to apply Lemma 4.15 on each block in K. Therefore, we construct iteratively an β_2-γ-α_1-path P of K such that no nontrivial outer P-bridge of K is part of an inductive call of Theorem 2.1. By Lemma 4.5 we can modify Q such that the union of P, Q, α and β forms the desired Tutte cycle that contains α, β and γ. □

4.5 A Constructive Proof for Sanders's Theorem

At this point we are ready to prove Theorem 2.2 constructively. We mostly follow the proof given in [59] but use the Three Edge Lemma (Lemma 4.16) and the tools developed for the constructive proof of Theorem 2.1 at crucial points to avoid constructing overlapping subgraphs. As before, we consider a 2-connected plane graph G and the proof will be by induction on the number of vertices in G (again, the base case is the triangle-graph, for which the claim can easily be verified). Together with G we are given vertices x and y both not in C_G and an edge $\alpha \in C_G$. If in the induction step either x or y are in C_G, then the claim follows directly from Theorem 2.1. Therefore, from here on we may assume that x and y are not in C_G. In addition, if G is decomposable into G_L and G_R, Theorem 2.2 holds by Lemma 4.1. The same holds if Lemma 4.2 can be applied to G. Therefore, we assume that G is non-decomposable. If there is an edge $e \in E(G)$ such that at least one of x or y is contained in C_{G-e}, the assumption that Lemma 4.1 and Lemma 4.2 are both not applicable on G imply that $G-e$ is 2-connected. This, in turn, allows us to construct an x-α-y-path of $G-e$ (and thus of G) by applying Theorem 2.1 on $G-e$. Thus, we assume no such edge e exists.

The previous observations show that there is no 2-separator in G that has both vertices in C_G and separates x and y. Hence, x and y are in the same component of $G-C_G$. Let K be the minimal plane chain of blocks $B_1, B_2, \dots, B_l$ in $G-C_G$ such that $x \in B_1$ and $y \in B_l$. Let $v_i := B_i \cap B_{i+1}$ for every $1 \leq i \leq l-1$ and set $v_0 := x$ and $v_l := y$.

Let J be any $(K \cup C_G)$-bridge. In our proof for Theorem 2.1, we chose the vertex $r_\alpha \in K \cap C_G$ as *reference vertex* in order to define $C_G(J)$ consistently. Here, the situation is more complicated, as K and C_G are disjoint and thus no vertex in $K \cap C_G$ exists. Instead, we use any vertex $s \in C_G$ as *reference vertex*, which shares a face with some vertex of K (note that this may not be true for all vertices in C_G). Now let $C_G(J)$ be the shortest path in C_G that contains all vertices in $J \cap C_G$ and does not contain s as an inner vertex.

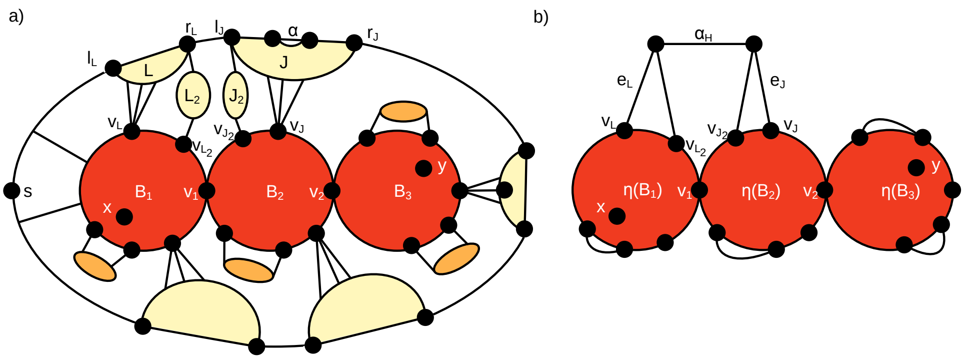


Figure 4.8: a) Decomposing G when both x and y are not in C_G. Here K consists of 3 blocks, $K_I = B_1 \cup B_2$ and $\mathcal{L}, \mathcal{J}$ are both of cardinality two. b) Shows the resulting $\eta(H)$ for the example in a).

By Lemma 4.2, Theorem 2.2 holds if there is a $(K \cup C_G)$-bridge of G whose attachments are all in C_G. Therefore, we assume that any $(K \cup C_G)$-bridge J of G has exactly one attachment in K and at least one attachment in C_G. Further, there must exist at least two $(K \cup C_G)$-bridges of G (although they might all be trivial), as $K \cap C_G = \emptyset$, and G is 2-connected. Let J be either the $(K \cup C_G)$-bridge for which $C_G(J)$ contains α, or, if no such bridge exists, the $(K \cup C_G)$-bridge for which l_J lies the closest counterclockwise to α on C_G (see Figure 4.8). Let L be the $(K \cup C_G)$-bridge for which r_L lies the closest counterclockwise to l_J on C_G (possibly $r_L = l_J$) such that $l_L \neq l_J$. Let $\mathcal{J} := \{J_1, J_2, \dots, J_p\}$ be the set of all $(K \cup C_G)$-bridges J_i for which $l_{J_i} = l_J$. Let $\mathcal{L} := \{L_1, L_2, \dots, L_q\}$ be the set of all $(K \cup C_G)$-bridges L_j for which $r_{L_j} = r_L$ and $l_{L_j} \neq l_J$. Then $J = J_i$ for some i and since $l_L \neq l_J$, $L \in \mathcal{L}$; hence, both $\mathcal{L}$ and $\mathcal{J}$ are non-empty. For any bridge $L_j \in \mathcal{L}$ we denote by v_{L_j} its unique attachment on K, and use a similar notation for the bridges in

$\mathcal{J}$. Let I be the minimal set of consecutive indices in $\{1, \ldots, l\}$ such that $K_I := \bigcup_{i \in I} B_i$ contains all attachments in K of the $(K \cup C_G)$-bridges in $\mathcal{L} \cup \mathcal{J}$. Let f and g denote the minimal and maximal indices of I.

To construct the desired x-α-y-path of G, we want to use the same strategy as in the previous sections (this is, define Q as a subpath of C_G and compute a Tutte path P of K iteratively). For this proof, this is slightly more complicated as the Tutte path we want to compute has to leave and reenter K in order to contain α. Therefore, we have to do some more preparation before we can apply Lemma 4.15 and Lemma 4.5. For this purpose we construct a second plane chain of blocks H from K and the $(K \cup C_G)$-bridges in $\mathcal{L} \cup \mathcal{J}$.

Initially, let H consist of K, two new artificial vertices a and b and the edge ab. For every $L_j \in \mathcal{L}$, we add an edge $e_{L_j} := v_{L_j} a$ to H (recall that v_{L_j} is the unique vertex $L_j \cap K$) and for every $J_i \in \mathcal{J}$, we add an edge $e_{J_i} := v_{J_i} b$ to H. We embed H on the plane by following the embedding of G and placing a and b into the outer face. If $r_L \neq l_J$, we are done with the construction of H and set $\alpha_H := ab$. Otherwise, we contract the edge ab of H and set $\alpha_H := v_{J_1} b$ (note that in this case $q = 1$). In both cases, H is a plane chain of blocks such that one block H_I contains the subgraph of G induced by the vertices in $K_I, \mathcal{L}$ and $\mathcal{J}$. Any other block in H is equivalent to some block B_i of K with $i < f$ or $i > g$. For every block B_i other than B_I, let α_i be an arbitrary edge of C_{B_i}.

If H consists of at least two blocks, then we define the boundary points for each block in H as follows:

- **For B_i, when $1 < i < f$ or $g < i < l$:** Let the boundary points of B_i be the vertices $v_{i-1}, l_{\alpha_i}, r_{\alpha_i}, v_i$ and let the boundary parts of B_i be the inclusion-wise maximal paths of C_{B_i} that do not contain any boundary point as inner vertex.

- **For B_1 and B_l if they are not contained in B_I:** If v_l is in C_{B_l} we define the boundary points of B_l and parts in the same way as in the previous case. The same holds for B_1 if $v_1 \in C_{B_1}$. As in our proof of Theorem 2.1, the case where $v_l \notin C_{B_l}$ depends on whether there exists a 2-separator in B_l that separators v_{l-1} and v_l. Therefore, consider for $v_l \notin C_{B_l}$ the 2-separator $\{w_1, w_p\} \subseteq C_{B_l}$ of B_l such that $B^+_{w_1,w_p}$ contains v_l, the path $w_1 C_{B_l} w_p$ is contained in one of the paths in $\{v_{l-1} C_{B_l} \alpha_l, \alpha_l, \alpha_l C_{B_l} v_{l-1}\}$ and $w_1 C_{B_l} w_p$ is of minimal length if such a 2-separator exists. If the 2-separator $\{w_1, w_p\}$ exists, let $w_1, \ldots, w_p$ be the $p \geq 2$ attachments of the $w_1 C_{B_l} w_p$-bridge of B_l that contains v_l, in the order of appearance in $w_1 C_{B_i} w_p$; otherwise, let for notational convenience $w_1 := \cdots := w_p := l_{\alpha_i}$. In the case $v_l \notin C_{B_l}$, let the boundary points of B_l be $v_{l-1}, l_{\alpha_l}, r_{\alpha_l}, w_1, \ldots, w_p$ and let the boundary parts of B_l be the inclusion-wise maximal paths of C_{B_l} that do not contain any boundary point as inner vertex. As it might happen that $v_1 \notin C_{B_1}$ we define the boundary points and parts symmetric to B_l if $v_l \notin C_{B_l}$.

- **For B_I:** Note that as at least one of v_{f-1} and v_g must be in C_{B_I} as H consists of at least two blocks. If both v_{f-1} and v_g are in C_{B_I}, then let the boundary points of B_I be the vertices $v_{f-1}, l_{\alpha_I}, r_{\alpha_I}, v_f, v_J, v_L$ and let the boundary parts of B_I be the inclusion-wise maximal paths of C_{B_I} that do not contain any boundary point as inner vertex. If v_{f-1} or v_g is not in C_{B_I}, we again look for a 2-separator $\{w_1, w_p\}$ as defined in the similar case for B_1 and B_l. We then set the boundary points for B_I as$v_{f-1}, l_{\alpha_I}, r_{\alpha_I}, v_L, v_J$ and $w_1, \ldots, w_p$ (if $\{w_1, w_p\}$ exists). Let the boundary parts of B_I be the inclusion-wise maximal paths of C_{B_I} that do not contain any boundary point as inner vertex.

This defines $\eta(H)$ and by Theorem 2.1 the blocks $B_1, \ldots, B_f, B_g, \ldots, B_l$ of $\eta(H)$ contain a $v_{i-1} - \alpha_{B_i} - v_i$-path $P^{\eta}_{B_i}$ and $\eta(B_I)$ has a $v_f - \alpha_H - v_g$-path $P^{\eta}_{B_I}$.

In the case that H itself is a block it might occur that both x and y are not in C_H. In this case, we cannot apply Theorem 2.1. We will, therefore, use induction for that case. Besides, there might exist a 2-separator in H that separates both x and y from α_H. We next show how to choose the boundary points in this case.

There might exists a 2-separator $\{u_1, u_q\} \subseteq C_H$ of H such that $H^+_{u_1,u_q}$ contains x, the path $u_1 C_H u_q$ is contained in one of the paths in $r_{\alpha_H} C_H l_{\alpha_H}$ and $u_1 C_H u_q$ is of minimal length. Similarly, there might exists a 2-separator $\{w_1, w_p\} \subseteq C_H$ of H such that $H^+_{w_1,w_p}$ contains y, the path $w_1 C_H w_p$ is contained in one of the paths in $r_{\alpha_H} C_H l_{\alpha_H}$ and $w_1 C_H w_p$ is of minimal length. It might happen that $\{u_1, u_q\} = \{w_1, w_p\}$, but it is impossible for the two 2-separators to interlace. Let the boundary points of H be v_1, v_l (if they are in C_H),$l_{\alpha_H}, r_{\alpha_H}$ and $u_1, \ldots, u_p, w_1, \ldots, w_p$ (if the 2-separators $\{u_1, u_q\}$ and $\{w_1, w_p\}$ exist) and let the boundary parts of H be the inclusion-wise maximal paths of C_H that do not contain any boundary point as inner vertex. This suffices to fulfill the definition of $\eta(H)$. If v_0 or v_l is in C_H, then we can apply Lemma 4.15 on H to construct a $v_0 - \alpha_H - v_l$-path P_H of H. If this is not the case, then we will use induction on $\eta(H)$ to construct an $x - \alpha_H - y$-path P^{η}_H in $\eta(H)$. Since C_G contains at least three vertices, it follows that $|V(\eta(H))| < |V(G)|$ and thus we can apply the induction in that way. Let P_H be the result of applying Lemma 4.15 on $\eta(H)$ and P^{η}_H.

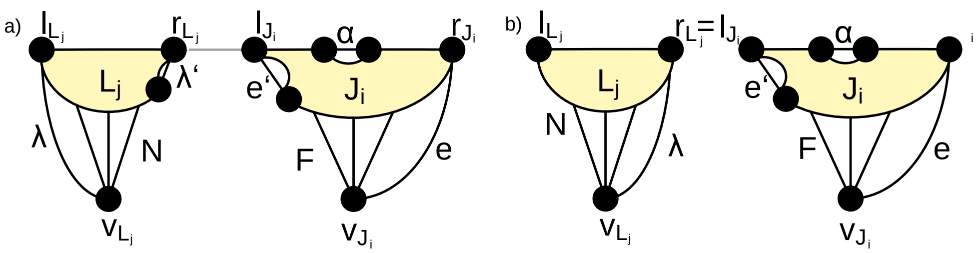


Figure 4.9: Two examples for the subgraphs N and F. In a) $r_{L_j} \neq l_{J_i}$, while in b) $r_{L_j} = l_{J_i}$.

So far P_H is not a subgraph of G, as it contains edges $v_{L_j} a$ and $v_{J_i} b$. Each of these edges represent a $(K \cup C_G)$-bridge of G. In the following, we show how to find Tutte paths P_{J_i} and P_{L_j} in L_j and J_i, respectively. Note that by forcing P_H through α_H we ensured that P_H contains exactly two of these artificial edges. If L_j or J_i are just single edges, let $P_{J_i} := J_i$ and $P_{L_j} := L_j$, respectively. If J_i is not just a single edge, let $e := v_{J_i} r_{J_i}$ and $F := J_i \cup C(J_i) \cup \{e\}$, where e is embedded such that $C(J_i)$ is part of the outer face of F. Let $e' \neq e$ be an edge in C_F incident to l_J (see Figure 4.9 for an example). Clearly, F is 2-connected and $|V(F)| < |V(G)|$. If $\alpha \in E(F)$ (i.e. $J_i = J$), then by Lemma 4.16 there is a Tutte cycle P' that contains e, e' and α. If $\alpha \notin J_i$, then by Theorem 2.1 there is a v_{J_i}-r_{J_i}-path P' in F through e'. In either case, let $P_{J_i} := P' - e$.

It remains to show what to do if L_j is not just an edge. If $r_{L_j} \neq l_J$, let $\lambda := v_{L_j} l_{L_j}$ and $N := L_j \cup C(L_j) \cup \{\lambda\}$, where λ is embedded such that $C(L_j)$ is part of the outer face of N. Let λ' be an incident edge to r_{L_j} that is different from λ of the outer face of N. Figure 4.9 shows an example for the construction of N. By Theorem 2.1 there is a v_{L_j}-λ'-l_{L_j}-path P_N of N. If otherwise $r_{L_j} = l_J$, then r_{L_j} is already part of P_{J_i} in J_i and we have to ensure that we do not include it as an internal vertex of P_N as well. Let $\lambda := v_{L_j} r_{L_j}$ and $N := L_j \cup C(L_j) \cup \{\lambda\}$, where λ is embedded such that C_{L_j} is part of the outer face of N. By Theorem 2.1, there is a l_{L_j}-λ-r_{L_j}-path P_N of N and we set $P_{L_j} := P_N - \lambda$. Note that if we consider the union of P_{L_j} and P_{J_i}, then any P_{L_j}-bridge in L_j that has r_{L_j} as an attachment will also have it as an attachment in $L_j \cup J_i$.

At this point we can remove a and b from P_H, note that this disconnects P_H. By adding P_{J_i} and P_{L_j} we end up with a path P_x from x to l_{L_j} and P_y from r_{J_i} to y. Let $Q := r_{J_i} C_G l_{L_j}$, to complete the proof of Theorem 2.2, we need to modify Q such that any $(P_x \cup P_y \cup Q)$-bridge of G has at most three attachments and exactly two if it contains an edge of C_G.

As G is such that Lemma 4.2 cannot be applied, there cannot be any $(P_x \cup Q \cup P_y)$-bridge with all its attachments in Q. Thus any $(P_x \cup Q \cup P_y)$-bridge of G has at least one attachment in P_x or P_y. At this point we want to apply the lemmas from Section 4.2 to Q. One of the prerequisites of that section is that Q and the given plane chain of blocks in G_Q are such that they cover all vertices in C_G. We can achieve this in the current setting by contracting all internal edges of the bridges in $\mathcal{L}$ and $\mathcal{J}$ to one of their attachments in C_G. We call the resulting graph G^*. Once we modified Q we will reverse this process, which does not change the number of attachments of any $(P_x \cup Q \cup P_y)$-bridge of G as the internal vertices of the bridges in $\mathcal{L}$ and $\mathcal{J}$ do not share any vertices other than their attachments with any subgraph that is touched during this step. As the process in Section 4.2 guarantees that Q is not changed in these vertices, adding these bridges back to G^* does not change the number of attachments of any $(P_x \cup Q \cup P_y)$ in G.

By Lemma 4.4, any $(P_x \cup Q \cup P_y)$ of G^* can have at most two attachments in $P_x \cup P_y$. By Lemma 4.5, every $(P_x \cup Q \cup P_y)$-bridge J of G^* contains a $l_J - r_J$-path Q_J. We replace $l_J C_G r_J$ in Q with Q_J for every such $(P_x \cup Q \cup P_y)$-bridge. Since l_J and r_J are contained in Q, no $(P_x \cup Q \cup P_y)$-bridge of G^* other than J is affected by this "local" replacement. Finally, after transforming G^* back to G, $P_x \cup Q \cup P_y$- is the desired $x - \alpha - y$-path of G.

4.6 A Quadratic Time Algorithm

In this section, we give an algorithm based on the decompositions shown in Chapter 4 (see Algorithm 4.1). Note that the description of Algorithm 4.1 only changes in the definition of K and Q when we want to compute either Theorem 2.1 or Theorem 2.2. It is well known that there are algorithms that compute the blocks of a graph and the block-cut tree of G in linear time, see [64] for a very simple one. Using this on $G - Q$ in either case, we can compute the blocks $B_1, \dots, B_l$ of K in time $O(n)$.

We now check if Lemma 4.1 or 4.2 is applicable at least once to G; if so, we stop and apply the construction of either Lemma 4.1 or 4.2. Checking applicability involves the computation of special 2-separators $\{c, d\}$ of G that are in C_G (e.g., we did assume minimality of $|V(G_R)|$ in Lemma 4.1). In order to find such a $\{c, d\}$ in time $O(n)$, we first compute the *weak dual* G^* of G, which is obtained from the dual of G by deleting its outer face vertex, and note that such pairs $\{c, d\}$ are exactly contained in the faces that correspond to 1-separators of G^*. Once more, these faces can be found by the block-cut tree of G^* in time $O(n)$ using the above algorithm. Since the block-cut tree is a tree, we can perform dynamic programming on all these 1-separators bottom-up the tree in linear total time, in order to find one desired $\{c, d\}$ that satisfies the respective constraints (e.g. minimizing $|V(G_R)|$, or separating x and α).

Now we compute $\eta(K)$. Since the boundary points of every B_i are known from K, all *maximal* 2-separators can be computed in time $O(n)$ by dynamic programming as described above. We compute the nested tree structure of all 2-separators on boundary parts due to Lemma 4.12, on which we then apply the induction described in Lemma 4.15. Hence, no nontrivial outer P-bridge of K is touched in the induction, which allows us to modify Q along the induction of Lemma 4.5.

Algorithm 4.1 TPATH(G, x, α, y) ▷ method, running time without induction

if G is a triangle or $\alpha = xy$ **then return** the trivial x-α-y path of G ▷ $O(1)$
if Lemma 4.1 or 4.2 is applicable at least once to G **then** ▷ weak dual block-cut tree, $O(n)$
    apply TPATH on G_L and G_R as described and **return** the resulting path ▷ $O(1)$
if there is a 2-separator $\{c, d\} \in C_G$ of G **then**
    do simple case 2
Compute the minimal plane chain K of blocks of G ▷ block-cut tree of $G - Q$, $O(n)$
Compute $\eta(K)$ ▷ dyn. progr. on weak dual block-cut tree, $O(n)$
Compute P by the induction of Lemma 4.15 ▷ dyn. progr. precomputes all possible B^+_{cd}, $O(n)$
Modify Q by the induction of Lemma 4.5 ▷ traversing outer faces of bridges, $O(n)$
return $P \cup \{\alpha\} \cup Q$

In our decomposition, every inductive call is invoked on a graph having fewer vertices than the current graph. The key insight is now to show a good bound on the total number of inductive calls to Theorem 2.2. To obtain good upper bounds, we will restrict the choice of α_i for every block B_i of K(which was almost arbitrary in the decomposition) such that α_i is an edge of $C_{B_i} - v_{i-1}v_i$. This prevents several situations in which the recursion stops because of the case $\alpha = xy$, which would unease the following arguments. The next lemma shows that only $O(n)$ inductive calls are performed. Its argument is, similarly to one in [18], based on a subtle summation of the Tutte path differences that occur in the recursion tree.

Lemma 4.17. *The number of inductive calls for TPATH*(G, x, α, y) *is at most* $2n - 3$.

Proof. Let r be the number of inductive calls for TPATH(G, x, α, y). Let $d(i)$, $1 \leq i \leq r$, be the number of smaller graphs into which we decompose the simple 2-connected plane graph of the ith inductive call. Let r' be the number of inductive calls that satisfy $d(i) = 1$. Let t be the number of graphs in which we can find the desired Tutte paths trivially without having to apply induction again (i.e., triangles or graphs in which $\alpha = xy$).

Thus, in the directed recursion tree, t is the number of leaves and r is the number of internal nodes, r' out of which have out-degree one. Since in a binary tree the number of internal nodes is one less than the number of leaves, the tree has at most $t - 1$ internal nodes of out-degree two or more. Thus we have

$$r \leq t - 1 + r'.$$

To complete the proof, we will give an upper bound for t that depends on n. The t instances in the leaves come in three different shapes: a triangle, a graph in which K consists of only one trivial block and Q can be found without applying induction (i.e., a cycle of length four) or a graph in which $\alpha = xy$. Any other instance is either decomposable into G_L and G_R or K contains at least one nontrivial block on which we have to apply induction. If the graph in a leaf instance is just a triangle the trivially found Tutte path will be of length two and we denote the number of such leaves by t_1. If a leaf represents a cycle of length four, then the trivially found Tutte path will be of length three. Let t_2 denote the number of such leaves. If the graph in the leaf instance is such that $\alpha = xy$, then the Tutte path returned for this instance will be of length one. Note that this case can only appear in the root instance. This follows from the fact that we always choose *alpha* such that $alpha \neq xy$ before we apply induction on a graph constructed in our decomposition. Thus if there is a leaf in which $alpha = xy$ then the tree consists of exactly one node and the claim is trivially true. Therefore, we assume that there is no such leaf from here on. Then there are $t = t_1 + t_2$ leaves and the sum over all paths lengths in the leaves is exactly $2t_1 + 3t_2$. In addition, a Tutte path in G has length at most $n - 1$. Combining these two facts, an upper bound on $2t_1 + 3t_2$ can be derived by going through every internal node of the recursion tree and adding the differences between the length of the Tutte path in the current node and the sum of lengths of the Tutte paths in its children nodes to $n - 1$.

If G is decomposable into G_L and G_R , then $d(i) = 2$ and the Tutte path P of G is either $(P_L \cup P_R) - cd$ or $(P_L - d) \cup (P_R - z)$. In the first case, P_L and P_R intersect in cd, and therefore $|E(P_L) + |E(P_R)| - |E(P)| = 1 = d(i) - 1$. In the latter case, P_L contains cd and P_R contains one edge incident to z, which both will not be part of P; therefore, $|E(P_L) + |E(P_R)| - |E(P)| = 2 = (d(i) - 1) + 1$.

Otherwise, the graph G of inductive call i is decomposed along certain 2-separators and $d(i)$ depends on the number of blocks in K, the number of such 2-separators and the resulting $(P \cup Q)$-bridges in G. The following argument will also hold for inductive calls when we apply Lemma 4.2, as the construction, is similar to the case when K consists of only one block and there is exactly one 2-separator in K. Note that only the inductive calls on the graphs split off from K increase the difference between the length of the Tutte path of G and the sum off Tutte path lengths found in the children of i, as only in this case the graphs in the parent node and its child overlap by one edge (the decomposition shows that this is the only possible overlap).

When constructing P using the induction of Lemma 4.15, we start with one inductive call for every block of $\eta(K)$. Every such block and every graph split off from K that needs an inductive call represents another child in the recursion tree. Initially, P is a Tutte path in $\eta(K)$ formed by the union of the Tutte paths $P_1^\eta, \ldots, P_l^\eta$, found in $\eta(B_1), \ldots, \eta((B_l)$, where l is the number of blocks in K. As P_j and P_{j+1}, $1 \leq j \leq l - 1$, do only intersect in one of their endvertices, the difference in $\sum_{j=1}^{l} |E(P_j)|$ and $|E(P = P_1 \cup \cdots \cup P_l)|$ is zero. For every graph that creates a child j that is split off from K, we remove one edge from P and replace it with a Tutte path P_j of j. As P and P_j do not intersect in any edge, $|E(P)| + |E(P_j)| - |E(P \cup P_j)| = 1$. Thus, the difference between the length of the Tutte path computed in i and the sum of lengths of Tutte paths computed in its children nodes is equal to the number k of graphs we split off from K and apply induction on. As $k \leq d(i) - 1$ the difference therefore is at most $d(i) - 1$ in this case.

If $d(i) = 1$, then the Tutte path found in the child note must be at least one edge shorter than the Tutte path in the parent node. Combining all of these differences shows that the total length of paths found in the t leaves is at most

$$\begin{aligned} 2t_1 + 3t_2 &\leq n - 1 + \sum_{1 \leq i \leq r} (d(i) - 1) + I - r' = n - 1 + r + t - 1 - r + I - r' \\ 2t + t_2 &\leq n + t + I - r' - 2, \end{aligned}$$

where I is the number of inductive calls on graphs that are decomposable into G_L and G_R. This implies that

$$\begin{aligned} t + t_2 &\leq n + I - r' - 2 \\ t &\leq n + I - r' - t_2 - 2 \leq n - r' + I - 2 \end{aligned}$$

Plugging this into the previous upper bound for r, we get $r \leq n + I - 3$. Note that no 2-separator can be used in more than one inductive call that decomposes the graph into G_L and G_R. Therefore, we obtain $I \leq n$ which concludes $r \leq 2n - 3$. □

Hence, Algorithm 4.1 has overall running time $O(n^2)$, which proves Theorem 2.8. We obtain as well the following direct corollary of the Three Edge Lemma 4.16.

Corollary 4.18. *Let G be a 2-connected plane graph and let α, β, γ be edges of C_G. Then a Tutte cycle of G that contains α, β and γ can be computed in time $O(n^2)$.*

CHAPTER 5

Conclusion

In this thesis, we have successfully refined the decomposition of circuit graphs into edge-disjoint subgraphs given in [60]. As a result, we were able to give an algorithm that computes a Tutte Path and from this a 2-walk in $O(n^2)$ and $O(n^3)$ time respectively. It remains open if there exists a way to bound the number of vertices visited twice in such a 2-walk and therefore make the computed structure even closer to a Hamiltonian Path of the given graph. The question of whether such a bound exists was risen in [53], where the authors show a similar bound for the number of degree three vertices in a 3-tree of the given circuit graph. As we can always construct a 3-tree from a 2-walk of the same graph, this question arises naturally.

We then showed that for both Thomassen's and Sanders's existence results on Tutte Paths in 2-connected planar graphs, there exists an $O(n^2)$ time algorithm that computes the promised Tutte paths. It remains open if the running time of our algorithm can be improved or if there exists a completely different algorithm to compute a Tutte path as promised in Theorem 2.2 in linear time. As evidenced by [5] finding more restricted Tutte paths can be done in $O(n)$ time even in 2-connected planar graphs. This question remains relevant as there are still new applications surfacing for Tutte paths and for some older applications we need an algorithm that can compute Theorem 2.2 efficiently, as Theorem 2.1 is not strong enough to derive them.

The key for our proof of Theorem 2.8 was the ability to identify additional vertices and edges of the given graph, which must be part of any Tutte path computed in it (other than the prescribed vertices). Here these were the vertices contained in maximal 2-separators of the blocks appearing in our decomposition. In the future we should aim for a better understanding on which vertices and edges must always be part of a Tutte path in a given planar graphs. A better understanding of which parts of the graph have to be contained in any Tutte path would not only open the door for new applications, but also improve our understanding on how Tutte paths can be constructed algorithmically. So far all known algorithms follow a divide and conquer strategy where for every constructed subgraph we only know that the resulting Tutte path will contain the prescribed vertices and edge. For everything in-between we more or less rely on a blackbox to choose what will be part of the final output.

Another direction for future research, related to longest cycles in 3-connected planar graphs, would be extending the notion of a System of Distinct Representatives. If the given graph is such that we can limit the size of every bridge of a given Tutte path, then the existence of an ordinary SDR already gives a guarantee of the existence of a cycle whose length depends on the size of the bridges. Thus if we show that for certain 3-connected planar graphs we can not only find a SDR but even a system of multiple representatives we will immediately get new bounds for longest cycles and paths in 3-connected planar graphs.

PART II

A New Approach for the Maximum Planar Subgraph Problem

This part is the result of a close collaboration with Parinya Chalermsook and Sumedha Uniyal. It is based on two articles, the first appeared in the proceedings of the *11th International Conference and Workshop on Algorithms (WALCOM'17)* [14]. The second article was published in the proceedings of the *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)* [15].

CHAPTER 6

Introduction to the Maximum Planar Subgraph Problem

In the MAXIMUM PLANAR SUBGRAPH PROBLEM (MPS) the objective is to compute a planar subgraph with a maximum number of edges from a given graph. This problem has proven to be useful in several real-world applications, including for instance, *architectural floor planning*, and *electronic circuit design.* Besides the practical applications, the problem is fundamentally interesting in theory as it has often been used as a subroutine in solving other basic graph drawing problems: Graph drawing problems generally ask for an embedding of a given graph with respect to some optimization criterion. To draw the graph, one often starts with a drawing of a planar subgraph and then adds the remaining edges, yet missing from the input graph, such that they satisfy the criterion. Naturally, if following this strategy, one would like to start with a planar subgraph that contains the maximum number of edges.

MPS is known to be NP-hard [49], therefore past research has been heavily focused on approximation algorithms. Călinescu et al. showed that MPS is actually APX-hard [10]. That the problem is hard to approximate might come as a surprise considering the following fact: By Euler's formula, any planar graph with n vertices can have at most $3n - 6$ edges. This means that simply outputting a spanning tree of the given graph immediately yields a $\frac{1}{3}$-approximation algorithm. In an effort to overcome this barrier, many heuristics were proposed [19, 9, 22], but even though these strategies were more involved than simply computing a spanning tree, none was able to give better than a $\frac{1}{3}$-approximation guarantee.

The breakthrough came when Călinescu et al. (implicitly) proposed to augment a spanning tree by *edge-disjoint* triangles. Adding one such triangle to a spanning tree gives one more edge to our MPS solution, so it is only natural to aim for adding as many triangles as possible. The authors showed that a simple algorithm based on greedily adding disjoint triangles achieves a $\frac{7}{18}$-approximation guarantee and also devised a $\frac{4}{9}$-approximation algorithm by first computing a maximum triangular cactus subgraph. The factor of $\frac{4}{9}$, however, is also shown to be the limit of this approach, as there exists a graph for which even a maximum triangular cactus subgraph contains only a $\frac{4}{9}$-fraction of the number of edges of an optimal solution for MPS. This approach was based on the work by Lovász [50] from 1980, where he initiated the study of $\beta(G)$ (sometimes referred to as the *cactus number* of G), the maximum value of the number of triangles in a cactus subgraph of G, and showed that it generalizes the Maximum Matching problem and can be reduced to linear matroid parity. This implies that the cactus number of any given graph is polynomial time computable. In fact, there are many efficient algorithms for matroid parity (both randomized and deterministic), e.g. [16, 51, 54, 28]. When we study $\beta(G)$, notice that a cactus subgraph that achieves the maximum value of $\beta(G)$

would only need to have cycles of length three (triangles).

Certain special cases of Mps also have received attention, partly due to their connection to extremal graph theory. For instance, [27] shows that the problem is APX-hard even in cubic graphs. In [45], Kühn et al. showed a structural result that when the graph is dense enough (i.e. has a large minimum vertex degree), then there is a triangulated planar subgraph that can be computed in polynomial time. Therefore, Mps is polynomial time solvable when the minimum vertex degree is large. The proof of this result relies on Szemerédi's Regularity Lemma.

More recently, [11] shows an approximation algorithm for weighted Mps in which we are given a weighted graph G, and the goal is to maximize the total weight of a planar subgraph of G. In [12] maximum series-parallel subgraphs are considered and a $\frac{7}{12}$-approximation algorithm is given. In combinatorial optimization, there are several problems closely related to Mps. For instance, finding a maximum series-parallel subgraph [12] or a maximum outer-planar graph [10], as well as the weighted variant of these problems [11]; these are the problems whose objectives are to maximize the number of edges. Perhaps the most famous extremal bound in the context of cactus is the min-max formula of Lovász [50] and a follow-up formula that is more illustrative in the context of cactus subgraphs [66]. All these formulas generalize the Tutte-Berge formula [3, 71] that has been used extensively both in research and curriculum.

Another related set of problems has the objectives of maximizing the number of vertices, instead of edges. In particular, in the maximum induced planar subgraphs (i.e. given a graph G, one aims at finding a set of vertices $S \subseteq V(G)$ such that $G[S]$ is planar while maximizing $|S|$.) This variant has been studied under a more generic name, called *maximum subgraph with hereditary property* [52, 48, 34]. This variant is unfortunately much harder to approximate: $\tilde{\Omega}(|V(G)|)$ (the term $\tilde{\Omega}$ hides asymptotically smaller factors) hard to approximate [37, 43]; in fact, the problems in this family do not even admit any FPT approximation algorithm [13], assuming the *gap exponential time hypothesis.*

6.1 Our Results

The state of the art techniques for solving Mps have, more or less, reached their limitations already twenty years ago. In this thesis, we introduce a new viewpoint that highlights the essence of the previously known algorithmic results. This allows us not only to give better explanations on previous results but also to suggest potential directions for breaking the long-standing $\frac{4}{9}$ barrier.

First, we quantify the connection between the number of triangular faces in a subgraph and its size as a solution for Mps by introducing a new optimization problem that we call Maximum Planar Triangles (Mpt): Given a graph G, compute a subgraph H and a embedding with a maximum number of triangular faces. We show, in particular, that a $\frac{1}{4}$-approximation for Mpt would immediately imply a $(\frac{1}{2} - O(1/n))$-approximation for Mps and that a $(\frac{1}{6} + \varepsilon)$-approximation algorithm would suffice for improving the best known approximation factor. Unlike the question of finding disjoint triangles, maximizing possibly overlapping triangles can be hard to compute, as we show that Mpt is NP-hard.

Since Mpt captures the previous approaches of finding triangular structures, many known algorithms for Mps can also be seen as algorithms for Mpt; this includes the greedy algorithm by Călinescu et al.[10] and those by Poranen [57] who proposed two greedy algorithms aiming at incorporating triangles that are not necessarily edge-disjoint (and conjectured that these two heuristics would achieve a $\frac{4}{9}$-approximation guarantee). In particular, we introduce a systematic study of a greedy framework, that we call Match-And-Merge. Roughly speaking, the algorithms in this class iteratively find isomorphic copies of "pattern graphs" and merge connected components in the so far computed subgraph until no pattern can be applied. The algorithm in this class can be concisely described by a set of *merging rules* and the iterations to apply them. This class of algorithms is relatively rich: All known greedy algorithms for Mps can be cast concisely in this framework and analyzed for their performance for Mpt. The analysis of this result is tight, i.e., we show examples of graphs for which these heuristics would not give better approximations for Mpt, than their proven upper bounds. Besides, we show that there is a simple Match-And-Merge algorithm that achieves a factor of $\frac{1}{11}$ for Mpt, therefore being the first greedy algorithm that performs better than $\frac{7}{18}$ for Mps.

Theorem 6.1. *There is a simple greedy $\frac{1}{11}$ and $\frac{13}{33}$-approximation for* Mpt *and* Mps *respectively.*

Despite not being able to break the $\frac{4}{9}$ barrier, our greedy algorithm sheds some light on how overlapping triangles can be of an advantage.

We then shift our attention away from greedy algorithms to the study of the extremal properties of $\beta(G)$. The $\frac{4}{9}$-approximation for Mps was achieved through an extremal bound of $\beta(G)$ when G is a planar graph. In particular, it was proven that $\beta(G) \geq \frac{1}{3}(n-2-t(G))$, where $n = |V(G)|$ and $t(G) = (3n-6) - |E(G)|$ (i.e., the number of edges missing from a triangulation of G). Our main result of this part is summarized in the following theorem.

Theorem 6.2. *Let G be a plane graph. Then $\beta(G) \geq \frac{1}{6} f_3(G)$ where $f_3(G)$ denotes the number of triangular faces in G.*

It is not hard to see that $f_3(G) \geq 2n - 4 - 2t(G)$, therefore Theorem 6.2 also implies the result of [10].

Corollary 6.3. $\beta(G) \geq \frac{1}{3}(n-2-t(G))$. *Hence, any polynomial time linear matroid parity algorithm gives a $\frac{4}{9}$-approximation for* Mps*.*

On the other hand, we show that the extremal bound provided in [10] alone is not sufficient to derive a approximation algorithm for Mpt. By invoking the shown connection between Mpt and Mps, Theorem 6.2 implies the following result for Mpt.

Corollary 6.4. *Any polynomial time algorithm for linear matroid parity gives a $\frac{1}{6}$ approximation for* Mpt*.*

Our result further highlights the extremal role of the cactus number in finding a dense planar structure, as illustrated by the fact that our bound on $\beta(G)$ is more "robust" to the change of objectives from Mps to Mpt. It allows us to reach the limit of approximation algorithms that linear matroid parity provides for both Mps and Mpt.

In addition, our work implies that local search arguments alone are sufficient to "almost" reach the best-known approximation results for both Mps and Mpt in the following sense: Matroid parity admits a PTAS via local search [46]. Therefore, combining this with our bound implies that local search arguments are sufficient to get us to a $\frac{4}{9} + \varepsilon$ approximation for Mps and a $\frac{1}{6} + \varepsilon$ approximation for Mpt. Therefore, this suggests that a local search strategy might be a promising candidate for such problems.

6.2 Preliminaries

Let $G = (V, E)$ be a graph. For any subset $S \subseteq V$, we use $G[S]$ to denote the induced subgraph of G on S. We denote by $V(G)$ and $E(G)$ the set of nodes and edges of G respectively. We denote by the length of a face in a plane graph, the number of edges in its boundary. Moreover, if G is a plane graph we use $f(G)$ to denote the number of faces of G and by $f_j(G)$ the number of faces of G with length j. Let $t(G)$ denote the number of edges necessary to turn G into a maximal plane graph. By Euler's formula it follows that $|E(G)| + t(G) = 3|V(G)| - 6$ and therefore $t(G)$ does not depend on the embedding of G. The following lemma was proven in [10].

Lemma 6.5. *[10] For any plane graph G, $f_3(G) \geq 2|V(G)| - 4 - 2t(G)$.*

As the number of faces in a graph is always at least the number of triangular faces in that graph, the next lemma follows trivially from Euler's formula, but it is crucial to show the connection between approximation algorithms for Mps and Mpt.

Lemma 6.6. *Let H be any connected subgraph of a connected plane graph. Then $|E(H)| \geq |V(H)| + f_3(H) - 2$.*

For our analysis of the different approaches to approximating Mpt we will often invoke the following simple lemma, which was proven in [10] as part of the analysis of greedy approximation algorithms for Mps. It relates the number of triangles to the number of vertices in each component of a cactus subgraph.

Lemma 6.7. *[10] Let X be a connected cactus graph, then we have $|V(X)| = 2p + 1$ where p is the number of triangles in X.*

For the remainder of this thesis, whenever we discuss Mps or Mpt on a graph G, we will denote by OPT_{mps} the number of edges in a maximum planar subgraph H of G, and by OPT_{mpt} the maximum number of triangular faces in a plane subgraph H' of G. As only triangular faces contribute to the solution of Mpt, all cactus subgraphs used in this thesis will be triangular cactus subgraphs and we will simply denote them by cactus subgraphs from hereon.

6.3 Hardness of Maximum Planar Triangles

In this section, we prove that Mpt is NP-hard, as a by-product we are able to simplify the NP-hardness proof for Mps by Liu and Geldmacher [49].

Theorem 6.8. Mpt *is NP-hard.*

Our reduction is from the Hamiltonian path problem in bipartite graphs. In [44], it is shown that the Hamiltonian cycle problem in bipartite graphs is NP-complete; it follows easily that the same holds for the Hamiltonian path problem.

Construction: Let G be an instance of the Hamiltonian path problem, i.e. G is a connected bipartite graph with n vertices. Note that G is triangle-free. Let G' be a copy of G, augmented with two vertices s and t, where s and t are both connected to every vertex in $V(G)$; we call the edges that connect vertices in G to $\{s,t\}$ *auxiliary edges*. More formally, $V(G') = V(G) \cup \{s,t\}$ and $E(G') = E(G) \cup \{(s,v) : v \in V(G)\} \cup \{(t,v), v \in V(G)\}$.

Analysis: We argue that there exists a spanning subgraph H of G' and an embedding ϕ_H of H with $2n-2$ triangular faces, if and only if G has a Hamiltonian path. First, assume that G has a Hamiltonian path P. We show how to construct a spanning subgraph H of G', that has an embedding ϕ_H with $2n-2$ triangular faces. Let $V(H) = V(P) \cup \{s,t\}$ and $E(H) = E(P) \cup \{(s,v) : v \in V(P)\} \cup \{(t,v) : v \in V(P)\}$. For ϕ_H simply embed P on the plane on a vertical line, placing s and t on the left and right side of the line respectively.

To prove the converse, now assume that there exists a spanning subgraph H of G' and an embedding ϕ_H of H with at least $2n-2$ triangular faces. Notice that each triangular face in H must be formed by an edge in $E(G)$ (called *supporting edge*) together with two auxiliary edges as G is triangle-free. Denote by $H' = H \setminus \{s,t\}$, which is a subgraph of G. We will show that there exists a Hamiltonian path in H' and therefore also in G.

Let E_s and E_t be the sets of edges in H' that support triangles formed with s and t in H respectively. Notice that the number of triangles in ϕ_H is $|E_s| + |E_t|$. We need the following structural lemma.

Lemma 6.9. *The subgraph $(V(G), E_s)$ (respectively $(V(G), E_t)$) of H' has the following properties:*

i The maximum degree of a vertex in $(V(G), E_s)$ is at most two.

ii If $(V(G), E_s)$ contains a cycle C, then $E_s \setminus E(C) = \emptyset$.

Proof. We first prove (i). Assume otherwise that some vertex v is adjacent to three supporting edges vv_1, vv_2, vv_3 for s. Suppose that the triangular faces (s, v, v_1) and (s, v, v_2) are adjacent in ϕ_H, sharing the edge sv. Then the triangle (s, v, v_3) cannot be a face, as it must contain one of the two faces in $\{(s, v, v_1), (s, v, v_2)\}$, a contradiction.

For (ii) note that every edge in E_s is incident to at least one triangular face in H. Assume now that E_s contains a cycle C and $E_s \setminus E(C) \neq \emptyset$. As $E_s \subseteq E(H') \subseteq E(G)$ and G is bipartite $|V(C)| \geq 4$. Note that by planarity s and the edges in $E_s \setminus E(C)$ must be embedded on the same side of C (inside or outside of C). Once we embed C, s and all auxiliary edges between C and s, every edge in $E(C)$ is incident to a triangular face (one of which is the outer face of the current graph) formed with the auxiliary edges and the face on the other side of C. Embedding any edge of $E_s \setminus E(C)$ on the same side as s and adding the auxiliary edges from its endvertices to s results in destroying one of these triangular faces. □

Lemma 6.9 implies that all subgraphs in H' induced by the endvertices of supporting edges for s (or t) must either be a disjoint union of paths or a cycle. Therefore E_s and E_t contribute at most n edges each to the triangular faces in H. At the same time, we know that to form at least $2n-2$ triangular faces in ϕ_H, one of them must have size at least $n-1$. To complete the proof of Theorem 6.8 we consider the possible compositions of edges from E_s and E_t in H':

- If E_s or E_t induces a cycle C of length n, G contains a Hamiltonian path.
- If one of E_s and E_t has size at least $n-1$ and at the same time induces a single path in H', then this path is also a Hamiltonian path in G.
- It remains to analyze the case where both E_s and E_t induce a cycle of length $n-1$ in H'. Let C be the cycle induced by E_s in H' and u be the vertex in $V(G) \setminus V(C)$. As G is connected there is a vertex v in C that is a neighbor of u in G. Let P be a path starting in u and ending in one of the neighbors of v in C. Clearly, P is a Hamiltonian path in G.

6.4 From MPT to MPS

We now show that any approximation algorithm for MPT can also be used to approximate MPS. Let G be an input instance for MPS, and H be a planar subgraph of G that corresponds to an optimal solution for MPS in G. For simplicity we abbreviate $|E(H)|$ and $|V(H)|$ by m and n respectively. We can always write m in terms of $(1+\gamma)n$ for some $\gamma \geq 0$.

Theorem 6.10. *If there is a β-approximation algorithm for* MPT*, then there is* $\min(\frac{1}{2}, \frac{1}{3} + \frac{2\beta}{3} - O(\frac{1}{n}))$*-approximation algorithm for* MPS.

Proof. By Euler's formula, $m = 3n - 6 - t(H)$, so $t(H) = (2-\gamma)n - 6$. If we fix an embedding of H, then by Lemma 6.5, the number of triangular faces in H must be at least $2n - 4 - 2t(H) = 2n - 4 - 2(2-\gamma)n + 12 > (2\gamma - 2)n$, what implies that $\mathsf{OPT}_{mpt} \geq (2\gamma - 2)n$. This term is only meaningful when $\gamma \geq 1$, so we distinguish between the following two cases that would imply Theorem 6.10.

- If $\mathsf{OPT}_{mps} < 2n$: This implies that any spanning tree is a $\frac{1}{2}$-approximation algorithm.
- Otherwise if $\mathsf{OPT}_{mps} \geq 2n$, then $\gamma \in [1, 2]$ (notice that γ can never be more than 2) and as argued above there are at least $(2\gamma - 2)n$ triangular faces in H. Then if we run a β-approximation algorithm for MPT, we will get a plane subgraph H' of G with $f_3(H') \geq \beta(2\gamma - 2)n$. We may assume that H' is connected: Otherwise, one can always add arbitrary edges to connect components without affecting planarity. By Lemma 6.6, $|E(H')| \geq \beta(2\gamma - 2)n + n - 2 = (1 + \beta(2\gamma - 2))n - 2$. The worst approximation factor is obtained by the infimum of the following term:

  $$\inf_{\gamma \in [1,2]} \frac{1 + \beta(2\gamma - 2)}{1 + \gamma}.$$

  To analyze this infimum, we first write a function $g(\gamma) = \frac{1+\beta(2\gamma-2)}{1+\gamma}$. The derivative $\frac{dg}{d\gamma}$ can be written as $\frac{4\beta-1}{(1+\gamma)^2}$. As long as $\beta \in (0, 1/4]$, we have $\frac{dg}{d\gamma} < 0$, so this function is decreasing in γ. This means that the infimum is achieved at the maximum value of γ, i.e. at the boundary $\gamma = 2$. Plugging in $\gamma = 2$ gives the infimum as $\frac{1+2\beta}{3}$, leading to the approximation ratio of $\frac{1+2\beta}{3} - 2/n$, as desired.

□

6.5 On the Strength of our Extremal Bound

The integral part to derive the improved approximation ration for Mps in [10] was to show that for any connected planar graph $G = (V, E)$ with $n = |V|$ vertices and $|E| = 3n - 6 - t(G)$ edges, the following holds.

Theorem 6.11 ([10]). *If G is a connected planar graph with $n \geq 3$ vertices, then $\beta(G) \geq \frac{1}{3}(n - t(G) - 2)$.*

We can show that a simple observation, Euler's formula and Theorem 6.2 together imply Theorem 6.11. By Euler's formula, a triangulated planar graph with n vertices has exactly $2n - 4$ faces. As removing one edge from a planar graph merges exactly two of its faces, removing k edges can destroy at most $2k$ triangular faces. Therefore, for any connected planar graph G we can easily give a lower bound on $f_3(G)$ that depends on $t(G)$.

Lemma 6.12. *If G is a connected planar graph, then $f_3(G) \geq 2n - 4 - 2t(G)$.*

Our extremal bound from Theorem 6.2 says that for any connected planar graph G, $\beta(G) \geq \frac{1}{6} f_3(G)$. Combining this with Lemma 6.12 yields

$$6\beta(G) \geq f_3(G) \geq 2n - 4 - 2t(G),$$

and therefore the same bound on $\beta(G)$ for a connected planar graph G as Theorem 6.11.

One might wonder if the reverse is true as well, i.e., can we use Theorem 6.11 to connect $\beta(G)$ to the number of triangular faces in a planar graph (and in turn directly use it to approximate Mpt). To this end we construct a graph in which $\frac{1}{3}(n - t(G) - 2) \leq 0$, even though $f_3(G) = \Theta(n)$ and thereby show that Theorem 6.11 alone is not enough for this task. Let G be a connected planar graph with n vertices, where $\frac{n}{2}$ vertices form a triangulated planar subgraph. Let v be any of the three vertices on the outer-face of this triangulated structure. We embed the remaining $\frac{n}{2}$ vertices of G in the outer-face and for each such vertex we add an edge to G, which connects it to v (see Figure 6.1 for an illustration of this construction). By Euler's formula, the initial triangulated subgraph with $\frac{n}{2}$ vertices has $\frac{3n}{2} - 6$ edges. In the next step, we added $\frac{n}{2}$ edges to connect the remaining vertices to v. Thus the resulting graph has exactly $2n - 6$ edges. Clearly, $t(G)$ is n in this graph. Using Euler's formula again, we can derive that the number of triangular faces in G is $f_3(G) = 2(\frac{n}{2}) - 4 - 1 = n - 5$ (all triangular faces of this graph are part of the initial triangulated subgraph, where we destroyed one triangular face by embedding the $\frac{n}{2}$ remaining vertices).

To prove Theorem 6.2 we use local search arguments, which work as follows. Let G be a plane graph, and let $\mathcal{C}$ be a cactus subgraph of G whose triangles correspond to triangular faces of G. The local search operation t-swap tries to replace up to t triangles in $\mathcal{C}$ by triangular faces of G such that the resulting cactus subgraph contains at least one more triangle than before. To be more formal: If there exists a collection $X \subseteq \mathcal{C}$ of t edge-disjoint triangles and a collection Y of at least $t + 1$ edge-disjoint triangles in $G \setminus E(\mathcal{C})$ such that $(\mathcal{C} \setminus X) \cup Y$ is a cactus subgraph of G, than set $\mathcal{C} := (\mathcal{C} \setminus X) \cup Y$. A

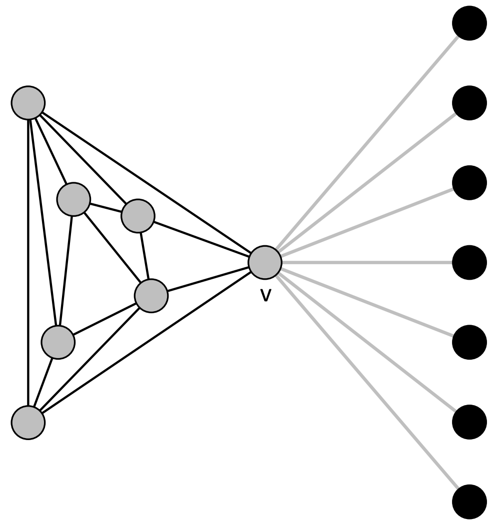


Figure 6.1: A graph that shows that an extremal bound as given by Theorem 6.11 for MPS does not necessarily imply a similarly strong result for MPT.

cactus subgraph is called t-swap optimal, if it can not be improved by a t-swap operation. An important point for the proof of Theorem 6.2 is that it suffices to only pick triangular faces from G as triangles in the computed cactus subgraph.

Using the gadget shown in Figure 6.2 one can illustrate the power of the local search arguments. Here, the triangular faces drawn with black solid lines form a 2-swap optimal cactus. By repeatedly adding copies of this gadget and merges them in the right way, one can construct an infinite family of graphs where any member G contains a 2-swap optimal cactus with at most $\frac{1}{6}f_3(G)$ triangular faces. This implies that the upper bound we show in Theorem 6.2 can already be met with a 2-swap optimal cactus subgraph. In fact, the gadget depicted in Figure 6.2 can also be slightly modified and then used to show that there exists an infinite family of graphs where for any member G there exists a 1-swap optimal cactus subgraph, that does not contain more than $\frac{1}{7}f_3(G)$ triangles.

We end this section by showing that there exists a graph G for which $\beta(G) \leq (\frac{1}{6} + o(1))f_3(G)$, therefore, showing that moving from a 2-swap optimal to a maximum cactus subgraph only improves the approximation factor slightly. A cactus subgraph C of a given graph G is called *maximal*, if there is no triangle T in $E(G) \setminus E(C)$ such that $C \cup T$ is again a cactus subgraph of G. We now show that a maximal cactus subgraph might only contain $\frac{1}{12}f_3(G)$ triangles for some connected planar graph G.

Lemma 6.13. *There is a family of n-vertex planar graphs $\{H_n\}_{n\in\mathbb{Z}}$ for which there exist a maximal cactus subgraph C_n of H_n such that* $\frac{f_3(C_n)}{f_3(H_n)} = \frac{1}{12} + o_n(1)$.

Proof. We start the construction of H_n with a cactus graph C_k consisting of k triangles that is embedded arbitrarily on the plane. We will augment C_k in two steps such that the resulting graph H_n will fulfill the claimed bound with respect to C_k. As this construction can be easily adapted for k and n growing to infinity, it will also describe an infinite

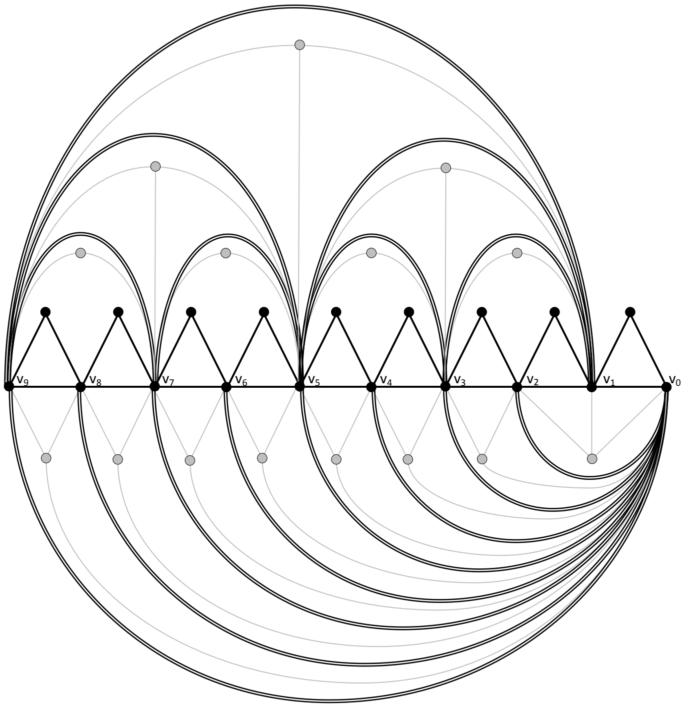


Figure 6.2: A gadget that can be used to construct a graph G where a 2-swap optimal cactus subgraph contains at most $\frac{1}{6}f_3(G)$ triangles.

family of graphs for which the claim will hold. First, we triangulate C_k by adding the necessary new edges and call the resulting graph C'. By Euler's formula C' has $2(2k+1)-4 = 4k-2$ faces. Then for each face of C' we add a vertex inside that face and connect it to the three vertices of the face boundary. The resulting graph H_n must, therefore, have $2k+1+4k-2 = 6k-1$ vertices, what we from hereon denote by n. Given n depending on k and using Euler's formula to determine the number of triangular faces in H_n, we get $f_3(H_n) = 2n-4 = 2(6k-1)-4 = 12k-6$. As $\lim_{n\to\infty} \frac{k}{12k-6} = \frac{1}{12}$ we can express $\frac{f_3(C_k)}{f_3(H_n)}$ by $\frac{1}{12} + o_n(1)$ for some constant that decreases if k (and therefore n) grows to infinity. □

We next show that in general a maximum cactus subgraph compared to a maximal cactus subgraph can have at most twice the number of triangles. This implies that even a maximum cactus subgraph can not have more than $\frac{1}{6} + o_n(1)$ triangles in H_n and thereby shows that Theorem 6.2 is tight up to a small constant.

Lemma 6.14. *Let G be a planar graph and let C be a maximal cactus subgraph of G. Then the number of triangles in C is at least $\frac{1}{2}\beta(G)$.*

Proof. Let C be a maximal cactus subgraph of G with k triangles. Let C^* be a maximum cactus subgraph of G with $\beta(G)$ triangles. We assume for contradiction that $\beta(G) \geq 2k+1$, then $|V(C^*)| \geq 2\beta(G)+1 \geq 4k+3$. Let V' denote the set of vertices in $V(C^*) \setminus V(C)$. From Lemma 6.7 it follows that $|V(C)| \geq 2k+1$, thus we can easily derive a lower bound on the number of vertices in V' as follows:

$$|V'| = |(V(C^*) \setminus V(C))| \geq |(V(C^*)| - |V(C))| \geq 4k + 3 - 2k - 1 = 2k + 2.$$

Note that any triangle in C^* can contain at most one vertex of V'. Otherwise, there would exist a triangle in G that intersects C in at most one vertex and thus could be added to C to form a larger cactus subgraph of G, contradicting the maximality of C. Therefore, for every vertex in V' there must exist one triangle in C^* that contains one vertex of V' and two vertices of $V(C)$. Let $E' \subseteq E(C^*)$ be the set of edges in C^* that connect the two vertices in each of these triangles with one vertex in V'. As C^* is a cactus subgraph any edge in E' can only be incident to one triangle in C^*, and therefore the graph induced by E' in $V(C)$ must be a forest. As this forest has exactly $2k+1$ vertices, there can be at most $2k$ triangles in C^* containing a vertex in V', contradicting that $|V'| \geq 2k + 2$ (as C^* is a cactus every vertex in $V(C^*)$ must be in some triangle). □

CHAPTER 7

Greedy Approximation Algorithms for MPT

We begin this chapter by formally introducing our Match-And-Merge framework for greedy algorithms for MPT. Afterwards, we analyze the approximation ratios of previously known algorithms for MPS in the context of MPT by rephrasing them in our new framework. In the final section of this chapter, we introduce a new greedy algorithm that outperforms all previously known greedy approximation algorithms for MPS. As discussed earlier, the MPT abstraction allows a cleaner analysis for algorithms in our framework, and therefore from hereon, we will focus on the case of MPT instead of MPS.

7.1 Match-And-Merge

To achieve a $\frac{4}{9}$-approximation for MPS in [10] the authors reduced MPS to the linear matroid parity problem. The reduction is constructive except for the process of picking the triangles for the final solution, which is done by the black-box that solves the linear matroid parity problem. We introduce a class of simple greedy algorithms so that we can focus on studying the advantage of picking (potentially) overlapping triangles.

First, we formally define the term *merging rules*. Let G be an input graph. At any point of execution of the algorithm, let E' be a subset of edges in $E(G)$ that have been included so far and $\mathcal{C}$ be the connected components in $G' = (V(G), E')$. Let H be a graph (that we refer to as pattern) and $\mathcal{P} = (V_1, V_2, \ldots, V_k)$ be a partition of $V(H)$. We say that an $(H, \mathcal{P})$-rule applies to G' if there is a subgraph H' in G that is isomorphic to H and such that, if we break H' into components based on $\mathcal{C}$ to obtain $U_1, \ldots, U_\ell$, then $\ell = k$ and $H'[U_i]$ is isomorphic to $H[V_i]$. When the rule is applied, all H-edges joining different components of $\mathcal{C}$ will be added. If $\mathcal{P}$ is a collection of singletons, we only use the abbreviation H-rule instead of $(H, \mathcal{P})$-rule: In this case, the rules would look for isomorphic copies of H where vertices come from different components in $\mathcal{C}$. Next, we will show how previously proposed algorithms fit into this framework. These algorithms are referred to as CA_0, CA_1 and CA_2 respectively [1].

- K_3**-rule**: The K_3-rule, when applied to G', will merge three connected components $C_1, C_2, C_3 \in \mathcal{C}$ such that there are $v_1 \in C_1, v_2 \in C_2, v_3 \in C_3$ where $\{v_1, v_2, v_3\}$ induces K_3. This rule has been used in many algorithms. The CA_0 algorithm in [12] can be concisely described in our framework as follows: Iteratively apply K_3-rule until it cannot be applied any further.

[1] In [57], Călinescu et al.'s algorithm was called CA, we change the name here to make it consistent with the names of the other algorithms in this thesis.

- **Poranen's rule:** The $(K_3, \{\{1,2\},\{3\}\})$-rule would look for a triangle (v_1, v_2, v_3) such that an edge (v_1, v_2) belongs to one component $C_1 \in \mathcal{C}$ and vertex v_3 to another component $C_2 \in \mathcal{C}$. The purpose of this rule is obvious: It will create triangles that are not necessarily disjoint. This rule has been used in two algorithms, CA_1 and CA_2, suggested by Poranen [57]. Both CA_1 and CA_2 use the same set of rules, except that they differ in the conditions on which the rule is applied. Lemma 7.3 shows that having more rules does not necessarily improve the performance of a greedy algorithm as CA_1 and CA_2 are proven to have the same lower bound.

7.2 Analyzing Previous Algorithms in our Framework

The first algorithm (called CA_0) we analyze for its performance in Mpt was introduced in [10] as the first algorithm to exceed the trivial $\frac{1}{3}$-approximation ratio for Mps. CA_0 can be phrased in the Match-And-Merge framework as follows:

(1) Repeatedly apply the K_3-rule until it cannot be applied anymore.

As this strategy does not guarantee more than that the resulting cactus subgraph is maximal in G, we can assume that the cactus subgraph constructed by CA_0 on one of the graphs H_n shown in Lemma 6.13 is exactly the cactus C_k. Therefore the approximation guarantee of CA_0 can not exceed $\frac{1}{12}$. In the following lemma we give a matching lower bound for this.

Lemma 7.1. *The approximation ratio of Algorithm CA_0 for* Mpt *is $\frac{1}{12}$.*

Proof. Let $H \subseteq E(G)$ denote the planar subgraph that CA_0 computed after the K_3-rule stops applying. Any component in H is either a collection of triangular faces or just a single vertex. Let $\mathcal{C} = \{C_1, \ldots, C_r\}$ be a collection of all components in H that contain at least one triangular face. Let p_i be the number of triangular faces found in component C_i, and p be the number of triangular faces found in S_1, so $p = \sum_{i=1}^{r} p_i$.

Let G^* be an optimal solution for Mpt in G and G_i^* the plane subgraph of G^* induced on C_i. It is easy to make the following observation.

Observation 7.2. *No triangle in G^* joins three different components of $\mathcal{C}$.*

Let $\Delta_{in}(C_i)$ denote the number of triangular faces in G^* that have all three vertices in $V(C_i)$ and let $\Delta_{out}(C_i)$ be the number of triangular faces in G^* with two vertices in $V(C_i)$ and a vertex not in $V(C_i)$. Then $\sum_{i=1}^{r}(\Delta_{in}(C_i) + \Delta_{out}(C_i)) = f_3(G^*)$, due to Proposition 7.2. Now notice that,

$$\frac{p}{f_3(G^*)} = \frac{\sum_{i=1}^{r} p_i}{\sum_{i=1}^{r}(\Delta_{in}(C_i) + \Delta_{out}(C_i))} \geq \min_i \frac{p_i}{\Delta_{in}(C_i) + \Delta_{out}(C_i)}.$$

Therefore, it suffices to show locally that $\frac{p_i}{\Delta_{in}(C_i)+\Delta_{out}(C_i)} \geq 1/12$. Note that every edge in G_i^* can be incident to at most two triangular faces in G^*. By Euler's formula there are at most $3|V(G_i^*)| - 6$ edges in G_i^*. Therefore $\Delta_{in}(C_i) + \Delta_{out}(C_i) \leq 6|V(G_i^*)| - 12$. In addition, Lemma 6.7 implies that $|V(C_i)| = 2p_i + 1$ for all i. Thus, $\Delta_{in}(C_i) + \Delta_{out}(C_i) \leq 6|V(C_i)| - 12 = 12p_i + 6 - 12 = 12p_i - 6$, and $\frac{p_i}{\Delta_{in}(C_i)+\Delta_{out}(C_i)} \geq \frac{1}{12}$ for every i. □

We continue our study of greedy strategies for MPT with the algorithms CA_1 and CA_2 given in [57] by Poranen. CA_1 can easily be phrased in the (in MATCH-AND-MERGE framework):

(1) Check if $(K_3, \{\{1,2\}, \{3\}\})$-rule applies

(2) If not, check if K_3-rule applies.

(3) If at least one of the rules applies, go back to (1).

It is easy to see that the output of CA_1 will always contain a maximal cactus subgraph of G as a subgraph and therefore will always perform at least as good as CA_0 for MPT (i.e. at least $\frac{1}{12}$-approximation for MPT). The algorithm CA_2 is the same as CA_1 with the restriction, that the $(K_3, \{\{1,2\}, \{3\}\})$-rule will only be applied if the edge $\{1,2\}$ is part of at most one triangle so far. This small difference results in CA_2 possibly producing non-outerplanar subgraphs while CA_1 will always output an outerplanar subgraph of G. In practice this makes CA_2 perform better than CA_1, but as we will show here, it does not help to prove a better approximation guarantee in theory. Based on their empirical successes in the experiments performed in [57], the author conjectured that they can even reach a $\frac{4}{9}$-approximation ratio in MPS matching the currently best-known algorithm given in [10]; this would hint to a $\frac{1}{6}$-approximation for MPT. We give a bad example where both algorithms can be as bad as a $\frac{1}{12}$-approximation for MPT and a $\frac{7}{18}$-approximation for MPS.

Lemma 7.3. *There is a graph G such that running CA_1 or CA_2 on G may yield at most $\frac{1}{12}OPT_{mpt}$ triangular faces, and $\frac{7}{18}OPT_{mps}$ edges.*

Proof. Assume that CA_1 on some input graph G, through poor choices when applying the K_3-rule, never gets the opportunity to apply the $(K_3, \{\{1,2\}, \{3\}\})$-rule. Then the resulting subgraph S of G will be a collection of edge-disjoint triangles that form a maximal cactus subgraph of G. We may assume that S consists of exactly one component with $2k+4$ triangles where any triangle intersects at most two other triangles and for any three triangles the intersection is empty. It is easy to find an embedding of S such that every triangle is also a triangular face. Note that we can assume that S is not spanning over all vertices of G, we will use this fact to construct another subgraph of G where only a constant number of edges is missing for it to be a triangulation. Afterwards, we will use the vertices in $V \setminus V(S)$ to introduce even more triangular faces to this new subgraph. All of these modifications will be made such that they do not contradict the assumption that CA_1 was not able to apply the $(K_3, \{\{1,2\}, \{3\}\})$-rule at any point in time when constructing S. As this rule was not applied at all, G also serves as a lower bound for CA_2, as both algorithms only differ in the way they apply this specific rule. As G must have at least as many edges and vertices as the newly constructed subgraph, comparing its size to S will imply that CA_1 and CA_2 only give a $\frac{1}{12}$-approximation for MPT and a $\frac{7}{18}$-approximation for MPS.

For simplicity assume that S is embedded on the plane on two parallel horizontal lines, i.e., any vertex in which two triangles intersect is put on the bottom line and the vertices that are not part of an intersection are on the top line (this is illustrated in Figure 7.1). We denote the triangular faces in this embedding of S in the following way.

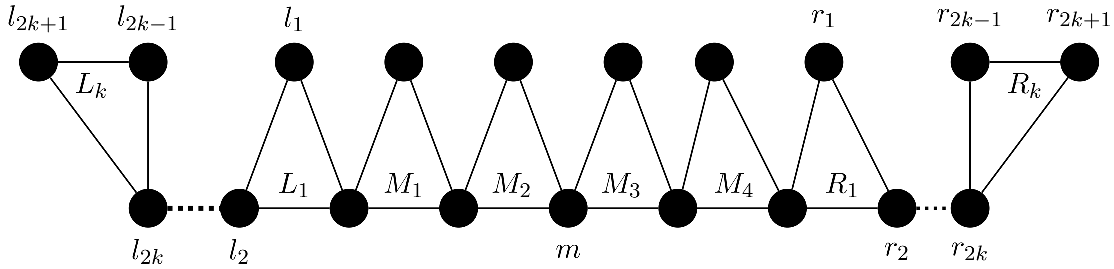


Figure 7.1: We may assume that CA_1 picks $2k + 4$ edge-disjoint triangles from G in a specific order.

Starting from the most left triangle we number the first k triangular faces from L_k to L_1. For $1 \leq i \leq k-1$ let l_{2i} denote the vertex in which L_i and L_{i+1} intersect and l_{2i-1} the vertex that does not intersect with another triangle. In L_k let l_{2k-1} and l_{2k+1} denote the two vertices that do not intersect with L_{k-1}. The four triangles to the right of L_1 play an important role in constructing the desired graph G, let them be denoted by M_1 to M_4 (from left to right) and denote the vertex in the intersection of M_2 and M_3 by m. We denote the remaining k triangles from left to right by R_1 to R_k and the vertices by r_1 to r_{2k} according to the same rules as done with the vertices in L_1 to L_k.

We assume that the algorithm picked the triangles S in a certain order. CA_1 started with picking the four triangles M_1, M_2, M_3, M_4 first and then L_1,L_2,...,L_k,R_1,R_2,..., R_k). As CA_1 could not apply the $(K_3, \{\{1,2\}, \{3\}\})$-rule, we know that there is no triangle in G that contains a vertex in $V \setminus V(S)$ and shares an edge with some triangle in S. Anyway, G can still contain many triangles other than M_2 and M_3 that contain m and intersect S in one edge. The algorithm was not able to use any of these triangles as in any point of time S consists of only one component and m is part of this component from the start. The same holds for triangles in G that contain an edge in S and l_1. In addition, there can still be many triangles left in G that contain two vertices that are in different triangles in S and contain a third vertex in $V \setminus V(S)$. We will use these three observations to construct our bad example subgraph G'.

Initially let $V(G') = V(G)$ and $E(G') = \{E(L_1) \cup E(M_1) \cup E(M_2) \cup E(M_3) \cup E(M_4)\}$. For i in $1, \ldots, 2k$ let $E(G') = E(G') \cup \{l_i r_i, l_1 r_i, m l_i, m r_i\}$. For $1 \leq i \leq 2k-1$ we connect r_i with l_{i+1} in G'. For i in $5, \ldots, 2k$ we additionally add $l_1 l_i$ to $E(G')$. Note that we cannot add such edges from l_1 to l_3 and l_4 as their existence in G would have allowed CA_1 to add triangles using the $(K_3, \{\{1,2\}, \{3\}\})$-rule, which it would have preferred over the assumed collection of triangles. We embed G' as follows in the plane. We place the vertices from $l_1, l_2, \ldots, l_k$ and $r_1, r_2, \ldots, r_k$ in an alternating order (i.e., $l_1, r_1, l_2, r_2, \ldots, l_k, r_k$) on a horizontal line and m somewhere above this line. Then we embed all edges that have one endvertex in r_i or l_i and the other in m as a straight line. The edges $l_1 l_5, l_1 l_6, \ldots, l_1 l_k$ and $l_1 r_2, l_1 r_3, \ldots, l1 r_k$ can be embedded as curves underneath the helper line. An embedding of G' as described above is shown in Figure 7.2.

Consider the subgraph H of G' induced by the vertices $l_1 \cup l_2 \cup \ldots l_{2k} \cup r_1 \cup r_2 \cup \ldots r_{2k} \cup m$. Let n' denote the number of vertices in H. We know that there are only three edges missing in H for it to be triangulated, therefore by Euler's formula H has $3n' - 6 - 3$ edges and $2n' - 4 - 6$ triangular faces (as every missing edge destroys two triangular faces), where

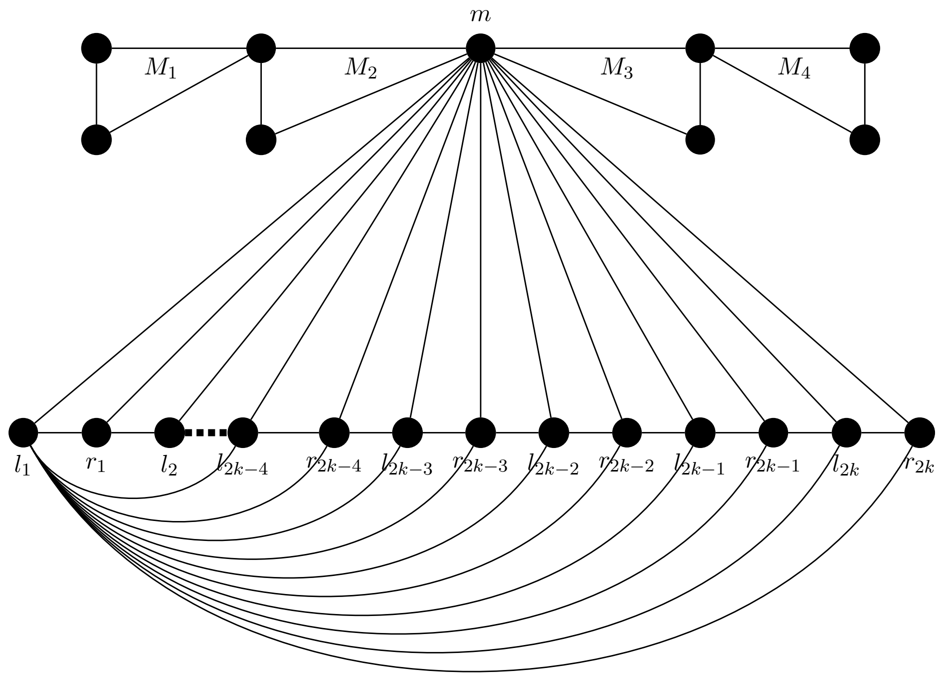


Figure 7.2: The subgraph G' of H induced by the vertices in S could look like this.

$n' = 4k+1$. Therefore H has $8k-8$ triangular faces. Assume that $|V \setminus V(S)| = 8k-8$ and put one of the vertices in $V \setminus V(S)$ into each of these faces. In addition, we add an edge from this vertex to each of the vertices in the face boundary. Clearly after this modification of H the number of triangular faces in the resulting graph is $3 \cdot (8k-8) = 24k-24$. If we also add M_1,M_2,M_3 and M_4 to H and this embedding and compare the number of triangles in H and S we get

$$\lim_{k\to\infty} \frac{2k+4}{24k-20} = \frac{1}{12}.$$

To see the performance of CA_1 in Mps we also have to consider the edges that the algorithm would have added in the next step to connect all vertices in $V \setminus V(S)$ to S. Therefore the number of edges in the graph constructed by $CA_1()$ in G would be $3\cdot(2k+4)+8k-8 = 14k+4$. Recall that H initially had $3n'-6-3 = 3\cdot(4k+1)-9 = 12k-6$ edges. For every triangular face, we then added one vertex and three more edges to H. Therefore $E(H) = 12k-6+3\cdot(8k-8) = 36k-30$. For increasing k we get an approximation ratio of

$$\lim_{k\to\infty} \frac{14k+4}{36k-30} = \frac{7}{18}.$$

□

The proofs of Lemma 7.1 and Lemma 7.3 conclude our studies of the previously best-known approximation algorithms for Mps and together gives us the following theorem.

Theorem 7.4. *The three algorithms CA_0, CA_1 and CA_2 are $\frac{1}{12}$-approximations for* Mpt.

7.3 A New Greedy Approximation Algorithm for MPS

We now propose a new rule that leads to a better approximation ratio. Let D_4 be the diamond graph (i.e. K_4 with one edge removed). This pattern graph intuitively captures the ideas of having two triangles sharing an edge. Our algorithm CA_3 proceeds in the following steps:

(1) Keep applying the D_4-rule until it cannot be applied any further.

(2) Keep applying the K_3-rule until it cannot be applied.

Now we analyze the performance of CA_3 in MPT. Let H be an optimal solution for MPT on a given graph G. Let $G' = (V, E')$ be the subgraph of G with E' as computed by CA_3 after leaving the second loop and $\mathcal{C} = \{C_1, \ldots, C_r\}$ be the collection of connected components in G'. Let $\mathcal{C}'$ be the connected components in G' formed after leaving the first loop; we call them *dense components.* (Notice that the components formed by diamonds are denser than those formed by adding triangles.) Notice that components in $\mathcal{C}$ are obtained by combining components in $\mathcal{C}'$. The fact that CA_3 can neither apply the D_4 nor the K_3-rule anymore implies that the following properties hold at the end of executing the algorithm.

Proposition 7.5.

- *For any four distinct dense components $X, Y, Z, W \in \mathcal{C}'$ and four vertices $x \in X, y \in Y, z \in Z, w \in W$, the induced subgraph $G[\{x, y, z, w\}]$ is not a diamond.*
- *For any three distinct components $X, Y, Z \in \mathcal{C}$ and three vertices $x \in X, y \in Y, z \in Z$, the induced subgraph $G[\{x, y, z\}]$ is not a triangle.*

For some connected component C in $\mathcal{C}$, we denote by $\Delta_{in}(C)$ the number of triangular faces in H whose three vertices belong to the induced subgraph $G[C]$. In addition, we denote by $\Delta_{out}(C)$ the number of triangular faces in H that have an edge in $G[C]$ and one vertex in $V \setminus V(C)$. The following lemma follows easily.

Lemma 7.6. $f_3(H) = \sum_{C \in \mathcal{C}} (\Delta_{in}(C) + \Delta_{out}(C))$.

Proof. Each triangular face $t = \{v_1, v_2, v_3\}$ of H such that v_1, v_2, v_3 belong to the same component is accounted for in $\sum_C \Delta_{in(C)}$. If two out of three vertices in t belong to the same component, triangle t is counted in the term $\sum_C \Delta_{out(C)}$. The remaining case when all vertices belong to different components cannot happen, due to Proposition 7.5. □

For a fixed component $C \in \mathcal{C}$, let $\Delta(C)$ denote the sum $\Delta_{in}(C) + \Delta_{out}(C)$.

$$\Delta(C) = \Delta_{in}(C) + \Delta_{out}(C) = (3\Delta_{in}(C) + \Delta_{out}(C)) - 2\Delta_{in}(C) \leq 2|E(H[C])| - 2\Delta_{in}(C).$$

The last inequality follows from the fact that each triangle contributing to $\Delta_{in}(C)$ uses three edges in C, while triangles in $\Delta_{out}(C)$ use only one edge.

Diamond clusters and triangular cacti: Fix some component C of $\mathcal{C}$. We can break C into several parts based on the structure of $\mathcal{C}'$. Let $\mathcal{D}_C$ be the collection of non-singleton dense connected components in C, i.e. $\mathcal{D}_C = \{C' \in \mathcal{C}' : C' \subseteq C \text{ and } |V(C')| > 1\}$. Each non-singleton subcomponent $X \in \mathcal{D}_C$ is called a *diamond cluster* inside C; notice that $|V(X)| \geq 4$. Let $F = E(G'[C]) \setminus \left(\bigcup_{X \in \mathcal{D}_C} E(G'[X])\right)$ be the edges remaining after removing edges in induced subgraphs of components in $\mathcal{D}_C$. Observe that the graph (C, F) consists of connected components that are formed by applying the K_3-rule. Let $\mathcal{T}_C$ be such a collection of non-singleton connected components. Each $Y \in \mathcal{T}_C$ is a connected triangular cactus in the component C. Notice that the components in $\mathcal{D}_C$ are disjoint, and the same holds for $\mathcal{T}_C$. For each $X \in \mathcal{D}_C$ and $Y \in \mathcal{T}_C$, let $c(X)$ and $l(Y)$ be the number of triangles in $G'[X]$ and that in $G'[Y]$ respectively.

Now we want to express the number of vertices $|V(C)|$ in terms of the sizes of the connected triangular cacti and diamond clusters in C. To simplify the following proofs we denote by p the number of triangles contained in diamonds of C and by l the number of non-diamond triangles in C (,i.e., $p = \sum_{X \in \mathcal{D}_C} c(X)$ and $l = \sum_{Y \in \mathcal{T}_C} l(Y)$).

Lemma 7.7. *The number of vertices in C can be written as*

$$|V(C)| = \frac{3}{2}p + 2l + 1.$$

Proof. We will show this by induction on the number of triangles in C. The equation is trivially true if $p = l = 0$. In the induction step, we take advantage of the treelike structure of C. By construction, any cycle in C has length at most four and is either a triangle or part of a diamond subgraph of G. This means that there always exists some triangles or diamonds in C that intersect other triangles or diamonds with at most one of their vertices. We call such triangles or diamonds the leaves of C. Note that if we take any leaf t from C and delete its vertices that do not intersect with other triangles or diamonds of C, then the resulting subgraph C' of C differs to C by either

(1) two vertices and one non-diamond triangle or

(2) three vertices and two diamond triangles.

Depending on whether t was a non-diamond or a diamond of C. If t is a non-diamond triangle, then by induction C' has $\frac{3}{2}p + 2(l-1) + 1 = \frac{3}{2}p + 2l - 1$ vertices. As C has only two vertices more than C' we get that $|V(C)| = \frac{3}{2}p + 2l + 1$. If t is a diamond of C, then by induction C' has $\frac{3}{2}(p-2) + 2l + 1 = \frac{3}{2}p - 3 + 2l - 1$ vertices. As C has exactly three vertices more than C' we get that $|V(C)| = \frac{3}{2}p + 2l + 1$. □

The following is the main lemma that crucially exploits the new diamond rule.

Lemma 7.8. $\Delta_{out}(C) \leq 15p + 8l - 6k$.

Proof. Each triangle that contributes to $\Delta_{out}(C)$ must have an edge that appears in $H[C]$; we call them *supporting edges*. Let E^* be the set of such edges. Denote by E_1^* the set of supporting edges whose two endvertices belong to the same diamond cluster $X \in \mathcal{D}_C$. Let E_2^* denote $E^* \setminus E_1^*$.

Claim 7.9. *The subgraph $(V(C), E_2^*)$ is triangle-free.*

Proof. Assume otherwise that there is a triangle (v_1, v_2, v_3) in (C, E_2^*). By definition of E_2^*, it must be the case that v_1, v_2 and v_3 must all lie in different diamond clusters. Moreover, since the edge (v_1, v_2) supports some triangle counted in $\Delta_{out}(C)$, we must have a vertex $v_4 \notin C$ such that $(v_1, v_4), (v_2, v_4) \in E(H)$. But then v_1, v_2, v_3, v_4 are joined by a diamond and belong to different components in $\mathcal{C}'$, contradicting Proposition 7.5. □

Now since $(V(C), E_2^*)$ is triangle-free, Euler's formula together with the upper bound on $|V(C)|$ imply that $|E_2^*| \leq 2|V(C)| - 4 \leq 3p + 4l - 2$. Moreover, we can bound the edges in E_1^* by applying Euler's formula to each diamond cluster $X \in \mathcal{D}_C$. That is, $|E_1^*| \leq \sum_{X\in\mathcal{D}_C}(3|V(X)| - 6) = \sum_{X\in\mathcal{D}_C}\left(\frac{9}{2}c(X) - 3\right) = \frac{9}{2}p - 3k$. Next, $\Delta_{out}(C) \leq 2|E^*|$ since each edge in E^* can only support at most two triangles. Plugging in the values of $|E_1^*|$ and $|E_2^*|$ gives

$$\Delta_{out}(C) \leq 2(|E_1^*| + |E_2^*|) \leq 2(\frac{9}{2}p - 3k + 3p + 4l - 2) \leq 15p + 8l - 6k.$$

□

We are now ready to prove the approximation guarantee of CA_3.

Lemma 7.10. *CA_3 gives a $\frac{1}{11}$-approximation for* MPT.

Proof. We will bound the approximation ratio locally, i.e. for each connected component C, we argue that $p+l \geq \frac{1}{11}\Delta(C)$, which will imply that when summing over all components in $\mathcal{C}$ the number of triangles is at least $\frac{1}{11}f_3(H)$. Using Euler's formula, we get

$$\Delta \leq 2|E(H[C])| - 2\Delta_{in} \leq 6|V(H[C])| - 12 - 2\Delta_{in} \leq 9p + 12l - 6 - 2\Delta_{in}. \quad (7.1)$$

The first inequality follows by a simple counting argument. Note that the last inequality follows from Lemma 7.7, which states that $|V(H[C])| \leq \frac{3}{2}p + 2l + 1$. From Lemma 7.8, we have that

$$\Delta = \Delta_{in} + \Delta_{out} \leq 15p + 8l - 6k + \Delta_{in}. \quad (7.2)$$

Adding (7.1) with twice of (7.2) gives us $3\Delta \leq 39p + 28l$, which implies that $\Delta \leq 13p + 10l$. Finally, we can combine this with (7.1) to get $\Delta \leq 11(p + l)$. □

CHAPTER 8

Computing the Number of Triangular Faces via Local Search

In this chapter, we show how a local search argument can be used to show that the cactus number for a given connected planar graph G is always at least one sixth of the triangular faces of G. For this, we start by explaining the necessary terminology and the key techniques we use in Section 8.1. In Section 8.2, we give a detailed overview of the proof by induction for Theorem 6.2. Section 8.4 focuses on showing the inductive argument and reducing the general case to proving the base case of the induction. In Section 8.4, we show a slightly weaker version of the base case that implies $\beta(G) \geq \frac{1}{7} f_3(G)$, and in Section 8.5, we prove the original base case to finally arrive at Theorem 6.2.

8.1 Taking Advantage of Local Optimality

Our proof for Theorem 6.2 is highly technical, although the basic idea is very simple and intuitive. Therefore, we first give a high-level overview of the analysis. Let $\mathcal{C}$ be a 2-swap optimal cactus subgraph of a given connected planar graph G. We argue that the number of triangles in $\mathcal{C}$ is at least $f_3(G)/6$. For simplicity, let us assume that $\mathcal{C}$ has only one non-singleton component. In general, one can repeat the following arguments for all other non-singleton components in $\mathcal{C}$. Let $S \subseteq V(G)$ be the vertices in this connected component.

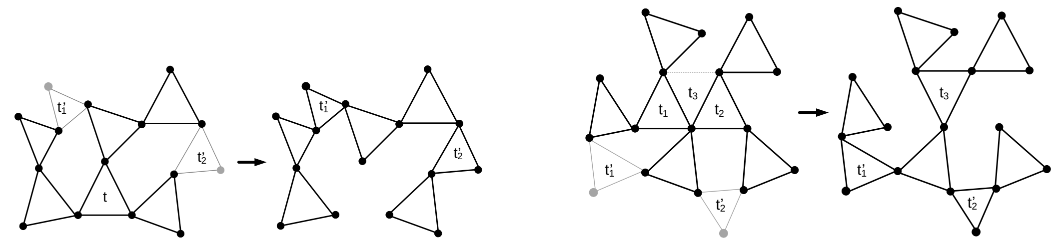
Let t be a triangle in $\mathcal{C}$. Notice that removing the three edges of t from $\mathcal{C}$ breaks the cactus subgraph into at most three components, say $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$ that are pairwise vertex-disjoint. Let $S_1.S_2$ and S_3 denote the vertex sets of $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$. Recall that we would like to upper bound the number of triangular faces in G by six times Δ, where Δ is the number of triangles in the cactus $\mathcal{C}$. Notice that $f_3(G)$ is comprised of $f_3(G[S_1]) + f_3(G[S_2]) + f_3(G[S_3]) + q'$, where q' is the number of triangular faces in G that span"across" the components S_1, S_2 and S_3 (i.e., those triangular faces whose vertices intersect with at least two sets S_i and S_j, where $i \neq j$). Therefore, if we could give a nice upper bound on q', e.g. if $q' \leq 6$, then we could inductively use $f_3(G[S_j]) \leq 6\Delta_j$, where Δ_j is the number of triangles in $\mathcal{C}_j$, to show that

$$f_3(G) \leq 6(\Delta_1 + \Delta_2 + \Delta_3) + 6 = 6(\Delta - 1) + 6 = 6\Delta.$$

and this would proof Theorem 6.2. Unfortunately, it is not possible to give such a nice upper bound on q' that holds in general for all situations. We will show, though, that such a bound can be proven for some suitable choices of t: Roughly speaking, removing such a triangle t from $\mathcal{C}$ will create only a small "interaction" between the components $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$ (i.e. small q'). We say that such a triangle t is a *light* triangle; otherwise, we say that it is *heavy*. As long as there is a light triangle left in $\mathcal{C}$, we would remove its edges from $\mathcal{C}$ (thus breaking $\mathcal{C}$ into $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$) and then use induction on each component. Therefore, we have reduced the problem to that of analyzing the base case of a cactus in which all triangles are heavy. Handling the base case of the inductive proof is the biggest challenge of our result.

We sketch here the two key ideas. First, we describe a way to exploit (in certain parts of the graph $G[S]$) that we are given a locally optimal solution. We want to point out; the fact that all triangles in $\mathcal{C}$ are heavy is crucial in this step. Recall that, each heavy triangle is such that its removal creates three components $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ with many "interactions" (i.e. many triangular faces of G span across these components) between them. However, intuitively, one would think that if there exist many triangles spanning across these components, then some of them could be used for making local improvements. Thus, the fact that there are many interactions will become our advantage in the local search analysis.

We briefly illustrate how we take advantage of heavy triangles. Let $\mathcal{T}$ be the set of triangular faces in G that are not contained in $\bigcup_i G[S_i]$, thus each triangle in $\mathcal{T}$ has vertices in at least two subsets S_j, S_i where $j \neq i$. The local search argument will allow us to say that all triangles in $\mathcal{T}$ have one vertex in S_i, one in S_j. with $i \neq j$, and one vertex not in $S_1 \cup S_2 \cup S_3$. This idea is illustrated in Figure 8.1(a). Moreover, we will even argue that there are not too many triangular faces in $G[S]$. One example of how to use a local search argument to show that certain types of triangular faces can not appear in $G[S]$ is illustrated in Figure 8.1(b).



(a) A 1-swap operation. If there exist two triangles t'_1 and t'_2 in $\mathcal{T}$ between two different pairs of components S_i, S_j (where $i \neq j$) of $\mathcal{C} \setminus E(t)$, then we cam remove t from $\mathcal{C}$ and add t'_1, t'_2 to construct a cactus subgraph with a larger number of triangles.

(b) A 2-swap operation. Let t_1 and t_2 be two adjacent triangles in $\mathcal{C}$. If there exists an edge between vertices in t_1 and t_2 (with distance two), and triangles t'_1 and t'_2 in $\mathcal{T}$ as drawn in this figure, then there exists a local improvement by removing t_1 and t_2 from $\mathcal{C}$ and adding t'_1, t'_2 and t_3 to $\mathcal{C}$.

Figure 8.1: Two examples which yield local improvements.

Finally, the ideas illustrated in both figures are only applied locally in a certain "region" inside the given connected planar graph G, therefore we still need a way to connect these regions to the number of triangular faces in all of G. Our final ingredient is a way to decompose the regions inside a plane graph into various "atomic" types. For each such atomic type, the local exchange argument is sufficient to argue about how close to optimality the number of triangles in a local optimal solution is compared to the number of triangular faces in that region in G. Combining the bounds on these atomic types gives us the desired result. This is the most technically involved part of this chapter, and we present it gradually by first showing the analysis that gives $\beta(G) \geq \frac{1}{7} f_3(G)$. For this, we need to classify the regions into five atomic types. To prove Theorem 6.2, that $\beta(G) \geq \frac{1}{6} f_3(G)$, we need a more complicated classification into thirteen atomic types.

8.2 How to Prove our Extremal Bound

In this section, we give a formal overview of the structure of the proof of Theorem 6.2. Let our input G be a plane graph and let $\mathcal{C}$ be a 2-swap optimal cactus subgraph of G. Let $\Delta(\mathcal{C})$ denote the number of triangles in $\mathcal{C}$, which correspond to triangular faces of G. We will show that $\Delta(\mathcal{C}) \geq f_3(G)/6$. In general, we will use the function $\Delta : G \to \mathbb{N}$ to denote the number of triangular faces in any plane graph G.

We partition the vertices in G into subsets based on the connected components of $\mathcal{C}$, i.e., $V(G) = \bigcup_i S_i$ where $\mathcal{C}[S_i]$ is a connected cactus subgraph of $\mathcal{C}$. For each i, where $|S_i| \geq 1$, let $q(S_i)$ denote the number of triangular faces in G with at least two vertices in S_i. The following proposition follows from the definition of $q(S_i)$ and the fact that $\mathcal{C}$ is a maximal cactus subgraph of G (which is implied by its 2-swap optimality). The proposition implies that $f_3(G) = \sum_i q(S_i)$.

Proposition 8.1. *If* $\Delta(\mathcal{C}_i) \geq \frac{1}{6} q(S_i)$ *for all* i*, then* $\Delta(\mathcal{C}) \geq \frac{1}{6} f_3(G)$*.*

Therefore, it is sufficient to analyze one component $\mathcal{C}[S_i]$ at a time, where $\mathcal{C}[S_i]$ contains at least one triangle (if the component does not contain at least one triangle it is just a single vertex) and show that $\Delta(\mathcal{C}_i) \geq \frac{1}{6} q(S_i)$. Thus, from now on, we fix one such component $\mathcal{C}[S_i]$ and denote S_i simply by S, $q(S_i)$ by $q(S)$, and $\Delta(\mathcal{C}[S_i])$ by p. We will show that $q \leq 6p$ through several steps.

Step 1: Reduction to Heavy Cactus

First, we will show that the general case can be reduced to the case where all triangles in $\mathcal{C}$ are *heavy* (to be defined below). We refer to different types of vertices, edges, and triangles in the graph G as follows:

- **Cactus:** All edges, vertices and triangles that are part of the cactus subgraph $\mathcal{C}[S]$ of G are called *cactus edges, -vertices and -triangles* respectively.

- **Cross:** Edges of G with one endvertex in S and one endvertex in $V(G) \setminus S$ are called *cross edges*. Triangles in G that contain one vertex from $V(G) \setminus S$ and two vertices from S are called *cross triangles*. Each cross triangle has exactly one edge in $G[S]$, we say this edge is the *supporting-edge* of this cross triangle. The component $\mathcal{C}[S_j]$ that contains the vertex v outside of S of a given cross triangle t is called the *landing component* of t. Similarly the vertex v alone is called the *landing vertex* of t.

- **type-i edges:** An edge in $G[S]$ that is not a cactus edge and does not support a cross triangle is called a *type*-0 *edge*. An edge in $G[S]$ that is not a cactus edge and supports i cross triangle(s) is called a *type-i edge*.

Therefore, each edge in $G[S]$ is either a cactus, type-0, type-1 or type-2 edge. The introduced naming convention makes it easier to make important observations like the following (see Figure 8.2 for an illustration of our naming convention).

Observation 8.2. *Triangles that contribute to the value of q are of the following types: (i) the cactus triangles; (ii) the cross triangles; and (iii) the "remaining" triangles that connect three cactus vertices using at least one type-*0*, type-*1 *or type-*2 *edge, and do not have a cross triangle embedded inside.*

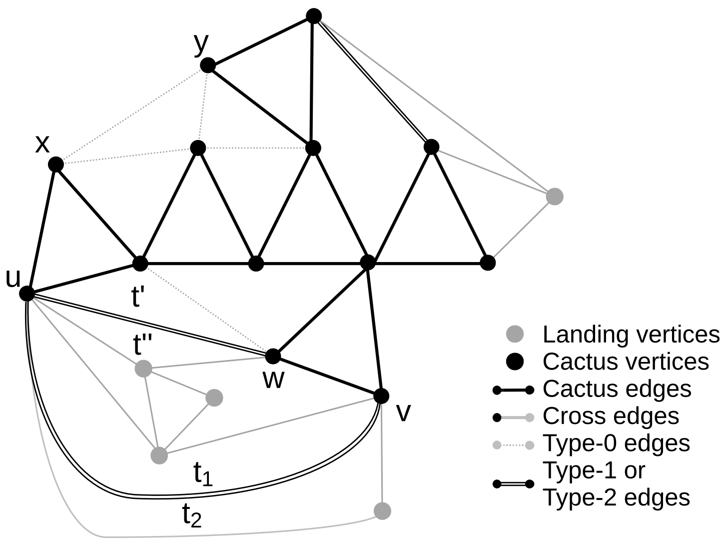


Figure 8.2: Various types of edges, vertices, and triangles. Here the cross triangles t'' and t_1 have the same landing component.

Using this classification of all the edges in $G[S]$, we can derive important information about the embedding of G and especially the landing components outside of $G[S]$.

Observation 8.3. *Any circuit C in G, which comprises of only cactus, type-*0*, type-*1 *and type-*2 *edges and cactus vertices, divides the plane into several regions (two if C is a cycle) such that any cross triangle which is embedded in one of the regions cannot share its landing component with any other cross triangle embedded in some different region.*

As for the edges in $G[S]$, we assign a type to every triangle in $G[S]$ and the cross triangles supported by edges in $G[S]$.

Types of cactus triangles and definition of split-cacti: Let t be a triangle in $\mathcal{C}[S]$. For $i \in \{0, 1, 2, 3\}$, we say that t is of type-i if exactly i of its edges support a cross triangle. Let p_i denote the number of type-i cactus triangles in $\mathcal{C}[S]$, thus we have that $p_0 + p_1 + p_2 + p_3 = p$. We denote the operation of deleting the edges of t from a connected cactus $\mathcal{C}[S]$ by *splitting* $\mathcal{C}[S]$ at t. The resulting three smaller triangular cacti (denoted by $\{\mathcal{C}_v^t\}_{v \in V(t)}$) are referred to as the *split-cacti* of t. For each $v \in V(t)$, let $S_v^t := V(\mathcal{C}_v^t)$ be the *split-components* containing v. For vertices $u, v \in V(t) : u \neq v$, we denote by B_{uv}^t the set of type-1 or type-2 edges having one endvertex in S_u^t and the other in S_v^t.

Using the different types of edges and triangles in G, we are finally ready to describe the concept of heavy and light cactus triangles, which will be heavily used in our analysis.

Heavy and light cactus triangles: We say that a cactus triangle t of $\mathcal{C}[S]$ is *heavy*, if either there are at least four cross triangles supported by edges in $E(t) \cup \bigcup_{uv \in E(t)} B^t_{uv}$ or there are at least three cross triangles supported by the edges in one set $B^t_{uv} \cup uv$ for some $uv \in E(t)$ and no cross triangle supported by the other sets $B^t_{ww'} \cup ww'$ for each $ww' \in E(t)$. Otherwise, t is *light*. Intuitively, the notion of a light cactus triangle t captures the fact that, after removing t, there is only a small amount of "interaction" between its split-components.

As another ingredient to bound the interaction in G between the components of $\mathcal{C}$, we define a function over the edges of the outer face of $\mathcal{C}[S]$.

Function ϕ: Denote by $\ell(S)$ the length of the outer-face f_S of the graph $G[S]$. We define $\phi(S)$ as the number of edges on the outer-face that do not support any cross triangles of G embedded in the outer-face of $\mathcal{C}[S]$, thus we have $0 \leq \phi(S) \leq \ell(S)$.

The main ingredients of Step 1 are encapsulated in the following theorem.

Theorem 8.4 (Reduction to heavy triangles)**.** *Let $\gamma \geq 6$ be a real number, and ϕ be as described above. If $q(S) \leq \gamma p(S) - \phi(S)$, for any connected component $\mathcal{C}[S[$ of $\mathcal{C}$ such that $\mathcal{C}[S]$ is a connected cactus subgraph of G that contains only heavy triangles, then $q(S) \leq \gamma p - \phi(S)$ for all connected components of $\mathcal{C}$.*

Therefore, it suffices to show the bound $q(S) \leq \gamma p - \phi(S)$ for the heavy cactus subgraphs of G. From this follows that $q \leq \gamma p$ in general (due to the non-negativity of the function ϕ). In other words, Theorem 8.4 gives a reduction from the general $\mathcal{C}[S]$ to the case when all cactus triangles in $\mathcal{C}[S]$ are heavy.

Step 2: The Skeleton Graph and Surviving Triangles

From hereon, we focus on the case when there are only heavy triangles in $\mathcal{C}[S]$ and we will give a formal overview of the key idea we use to derive the bound $q(S) \leq 6p - \phi(S)$. This bound in combination with Theorem 8.4 then implies Theorem 6.2.

Structural properties of heavy triangles: Using the local optimality of $\mathcal{C}$ one can show, that the light and heavy triangles in $\mathcal{C}$ behave in a very well structured manner. The following proposition summarizes these structural properties of heavy triangles (we delay the proof of this proposition to Subsection 8.2.1).

Proposition 8.5. *Let t be a cactus triangle in the cactus subgraph $\mathcal{C}[S]$ of G.*

- *If t is heavy, then t is either type-0 or type-1.*
- *If t is a heavy type-1 triangle, where the edge $uv \in E(t)$ supports the cross triangle supported by t, then $B^t_{ww'} = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$ and the total number of cross triangles supported by edges in B^t_{uv} is at least two.*

- *If t is a heavy type-0 triangle, then there is an edge $uv \in E(t)$ such that $B^t_{ww'} = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$ and the total number of cross triangles supported by edges in B^t_{uv} is at least three.*

By Proposition 8.5 there can only exist type-0 and type-1 cactus triangles in $\mathcal{C}[S]$. Moreover, for each such heavy cactus triangle t, the type-1 or type-2 edges in $G[S]$ only connect vertices of two split-components of t.

Skeleton graph H: Let a_i be the number of edges of type-i in $G[S]$. Notice that the number of non-cactus edges in $G[S]$ is exactly $\sum_i a_i = |E(G[S])| - 3p$. Let A be the set of all type-0 edges in $G[S]$. Let $H := G[S] \setminus A$ be a new graph that we call the *skeleton graph* of G. By definition H contains only cactus, type-1 or type-2 edges and every face f of H possibly contains multiple faces of G, thus we will refer to a face of H as a *super-face* of G. At high-level, we aim to analyze each super-face f and provide an upper bound on the number of triangular faces of G embedded inside f. Denote by $\mathcal{F}$ the set of all super-faces (except for the p faces corresponding to cactus triangles).

Let f be a super-face of H. We denote by $survive(f)$ the number of triangular faces of $G[S]$. embedded inside of f that do not contain any cross triangles in G. Next, we use a simple counting argument to derive q using the skeleton graph H based on three facts:

(1) There are p cactus triangles in $G[S]$.

(2) There are $p_1 + a_1 + 2a_2$ cross triangles supported by edges in $G[S]$.

(3) There are $\sum_{f \in \mathcal{F}} survive(f)$ triangular faces in $G[S]$ that were not counted in (1) or (2).

Combining these properties, we obtain:

$$q \leq p + (p_1 + a_1 + 2a_2) + \sum_{f \in \mathcal{F}} survive(f). \tag{8.1}$$

The first and second terms are expressed nicely in terms that describe the size of $G[S]$, thus the key is to achieve the best upper bound on the third term in terms of the same parameters. Roughly speaking, the intuition is the following: When a_2 or a_1 is high (meaning there are many edges in $G[S]$ supporting cross triangles), the second term becomes higher. However, each cross triangle needs to be embedded inside some super-face in H, therefore decreasing the value of the term $\sum_{f \in \mathcal{F}} survive(f)$. Similar arguments can be made for p_1. Therefore, the key to a tight analysis is to understand this trade-off and the structure of the super-faces of H.

The structure of super-faces: Let $f \in \mathcal{F}$ be a super-face of H. Recall that an edge in the boundary of f is either a type-1, type-2 or a cactus edge. We aim for a better understanding of the value of $survive(f)$. In general, this value can be as high as $|E(f)| - 2$, e.g. if the additional edges in $G[V(f)]$ are type-0 edges and such that $G[V(f)]$ is a triangulation of the region bounded by the super-face f. However, if some edge in the boundary of f supports a cross triangle whose landing component is embedded inside of f in G, then the possible value of $survive(f)$ decreases by one. So speaking, the edge supporting a cross triangle is *killing* the triangular face adjacent to it, hence the term *survive*. The following observation is crucial for our analysis:

Observation 8.6. *For some super-face f of H, consider any edge $e \in E(f)$. Then e is either*

- *of type-*1*, type-*2 *or a cactus edge and supports a cross triangle embedded in f or*
- *of type-*1*, type-*2 *or a cactus edge and does not support a cross triangle embedded in f.*

Edges lying in the first case of Observation 8.6 are called *occupied* edges (the set of such edges in $E(f)$ is denoted by $Occ(f)$). The edges in the boundary of f that are not occupied are called *free* edges in f (the set of free edges in $E(f)$ is denoted by $Free(f)$). By Observation 8.6, the number of edge in the boundary of f can be expressed by $|E(f)| = |Occ(f)| + |Free(f)|$. A very important quantity for our analysis is $\mu(f) = \frac{1}{2} \cdot |Occ(f)| + |Free(f)|$, which roughly bounds the value of $survive(f)$ (within some small constant additive terms).

We will assume without loss of generality that $survive(f)$ is the maximum possible value of surviving triangles that can be obtained by embedding type-0 edges in f, thus $\mu(f)$ is a function that depends only on the bounding edges in f. We define $gain(f) = \mu(f) - survive(f)$, which is again a function that only depends on bounding edges of f. Intuitively, the higher the term $gain(f)$, the better for us (since this would lower the value of $survive(f)$), and in fact, it will later become clear that $gain(f)$ roughly captures the "effectiveness" of a local exchange argument on the super-face f. Hence, it suffices to show that $\sum_{f \in \mathcal{F}} gain(f)$ is sufficiently large. The following proposition makes this precise:

Proposition 8.7. $\sum_{f \in \mathcal{F}} survive(f) = (3p - \frac{1}{2}p_1 + \frac{3}{2}a_1 + a_2) - \sum_{f \in \mathcal{F}} gain(f)$

Proof. Notice that $\sum_{f \in \mathcal{F}} \mu(f)$ can be analyzed as follows:

- Each cactus triangle is counted three times (once for each of its edges), and for a type-1 triangle, one of the three edges contribute only one half. Therefore, this accounts for the term $3p - \frac{1}{2}p_1$.
- Each type-1 or type-2 edge is counted two times (once per super-face containing it in its boundary). For a type-2 edge, the contribution is always half (since it always is accounted in $Occ(f)$). For a type-1 edge, the contribution is half on the occupied case, and full on the free case. Therefore, this accounts for the term $\frac{3}{2}a_1 + a_2$.

Overall we get, $\sum_{f \in \mathcal{F}} \mu(f) = 3p - \frac{1}{2}p_1 + \frac{3}{2}a_1 + a_2$, which finishes the proof. □

Combining this proposition with Equation 8.1, we get:

$$q \leq 4p + \frac{1}{2}p_1 + \frac{5}{2}a_1 + 3a_2 - \sum_{f \in \mathcal{F}} gain(f). \tag{8.2}$$

Using the gain function to prove a weaker bound on q: To recap, after Step 1 and Step 2, we have reduced the analysis to the question of lower bounding $\sum_{f\in\mathcal{F}} gain(f)$. We first illustrate that we could get a weaker (but nontrivial) result compared to Theorem 6.2 by using a generic upper bound on the gain function. In Step 3, we will show how to substantially improve this bound, allowing us to achieve the ratio of Theorem 6.2.

Lemma 8.8. *For any super-face (except for the outer-face) in $\mathcal{F}$, we have $gain(f) \geq \frac{3}{2}$.*

We denote by f_0 the outer (super-)face of H. As f_0 is special, we can achieve a lower bound on the quantity $gain(f_0)$ that depends on $\phi(S)$. This is captured by the following lemma, which we prove in subsection 8.2.2 at the end of this section.

Lemma 8.9. *For the outer-face f_0, we have that $gain(f) \geq \phi(S) - 1$.*

Combining Lemma 8.8 and Lemma 8.9 we get the following lower bound on the sum over all gain values of the super-faces in H.

$$\sum_{f\in\mathcal{F}} gain(f) \geq \phi(S) - 1 + \frac{3}{2}(|\mathcal{F}| - 1) = \phi(S) + \frac{3}{2}|\mathcal{F}| - \frac{1}{2}. \quad (8.3)$$

The following lemma upper bounds the number of skeleton faces (i.e. super-faces of the skeleton).

Lemma 8.10. $|\mathcal{F}| = a_1 + a_2 + 1 \leq 2p - 2$.

Proof. Proposition 8.5 allows us to modify the graph H into another simple planar graph $\widetilde{H}$ such that the claimed upper bound on $|\mathcal{F}|$ will follow simply from Euler's formula. Let t be a cactus triangle where $V(t) = \{u, v, w\}$ and $uw \in E(t)$ be such that the edge set B^t_{uw} is empty, as guaranteed in Proposition 8.5. For every cactus triangle t we contract the edge uw into one new vertex W. Note that this operation creates two parallel edges with endvertices W and v in the resulting graph. To avoid multi-edges in the resulting graph $\widetilde{H}$ we remove one of them (see Figure 8.3 for an illustration of this operation). Since B^t_{uw} is empty, this operation cannot create any other multi-edges in $\widetilde{H}$. In addition, the contraction of an edge maintains planarity, hence after each such transformation, the graph remains simple and planar. As a result of applying the above operation to all cactus triangles, the graph $\widetilde{H}$ has $p+1$ vertices and p edges corresponding to the contracted triangles. By Euler's formula the number of edges in $\widetilde{H}$ is at most $3(p+1) - 6 = 3p - 3$, which implies that $a_1 + a_2 \leq 2p - 3$, and as $|\mathcal{F}| = a_1 + a_2 + 1$ we get that $|\mathcal{F}| \leq 2p - 2$. □

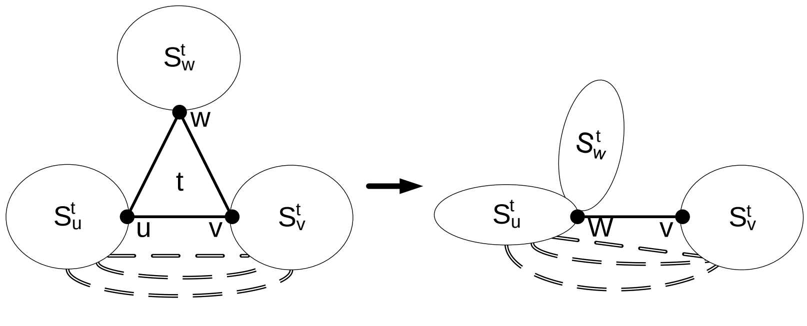


Figure 8.3: An example of the contraction transformation.

Combining the trivial gain (i.e. Inequality 8.3) with Inequality 8.2, we get

$$q \leq (4p+\frac{1}{2}p_1+\frac{5}{2}a_1+3a_2)-(\phi(S)+\frac{3}{2}(a_1+a_2+1)-\frac{5}{2}) = 4p+\frac{1}{2}p_1+a_1+\frac{3}{2}a_2-\phi(S)+1.$$

Now, using Lemma 8.10 and the trivial bound that $p_1 \leq p$, we get $q(S) \leq \frac{9}{2}p + \frac{3}{2}(a_1 + a_2) - \phi(S) + 1 \leq \frac{15}{2}p - \phi(S)$, therefore implying a factor $\frac{15}{2}$ upper bound.

Step 3: upper bounding Gain via Super-Face Classification

In this final step, we show another crucial ingredient on the way to reach the factor six ratio of Theorem 6.2. Intuitively, the most difficult part of lower bounding the total gain is the fact that the value of $gain(f)$ varies, depending on the composition of each super-face in H, and we cannot expect a strong "universal" bound that holds for all cases. For instance, Figure 8.4 shows a super-face with $gain(f) = \frac{3}{2}$, thus strictly speaking, we cannot improve the generic bound of $\frac{3}{2}$. This is why we now introduce a *classification scheme* for the super-faces in H. The goal here is to partition the super-faces in $\mathcal{F}$ into several types, such that all super-faces of one type have the same gain.

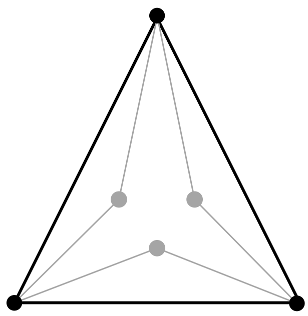


Figure 8.4: A super-face $f \in \mathcal{F}$ having $gain(f) = \frac{3}{2}$ as $\mu(f) = \frac{3}{2}$ and $survive(f) = 0$.

Super-face classification scheme: We aim to define a set of rules Φ that classify $\mathcal{F}$ into a fixed number of types. We say that the set of rules Φ is a d-type classification if the rules classify $\mathcal{F}$ into d sets $\mathcal{F} = \bigcup_{j=1}^{d} \mathcal{F}[j]$. Let $\vec{\chi}$ be a vector such that $\vec{\chi}[i] = |\mathcal{F}[i]|$. For each such set, we will prove a lower bound on the sum over all gain values of the contained super-faces. We define the gain vector by $\overrightarrow{gain}$ where $\overrightarrow{gain}[i] = \min_{f \in \mathcal{F}[i]} gain(f)$. The total gain can be rewritten as:

$$\sum_{f \in \mathcal{F}} gain(f) = \overrightarrow{gain} \cdot \vec{\chi}.$$

Notice that, the total gain value $\overrightarrow{gain} \cdot \vec{\chi}$ is written in terms of the $\vec{\chi}[j]$ variables, thus we need another ingredient to lower bound this in with respect to the variables p, p_1, a_1 and a_2. Therefore, another component of the classification scheme is a set of *valid linear inequalities* Ψ of the form $\sum_{j=1}^{d} C_j \vec{\chi}[j] \leq \sum_{j \in \{0,1\}} d_j p_j + \sum_{j \in \{1,2\}} d'_j a_j$. This set of inequalities will allow us to map the formula in terms of $\vec{\chi}[j]$ into one with respect to the variables p, p_1, a_1 and a_2.

A classification scheme is defined as a pair (Φ, Ψ). We say that such a scheme certifies the proof of factor γ if it can be used to derive $q(S) \leq \gamma p - \phi(S)$. Given a fixed classification scheme and a gain vector, we can check whether it certifies a factor γ by using an LP solver (although in our proof, we will show this derivation).

For the proof of Theorem 6.2 we will present a classification scheme that certifies a factor six. Since the proof is very complicated, we also provide a simpler, more intuitive proof that certifies a factor seven first.

Theorem 8.11. *There is a* 5*-type classification scheme, such that* $q(S) \leq 7p - \phi(S)$*.*

We remark that the analysis of factor seven only requires a cactus subgraph of G that is 1-swap optimal.

Theorem 8.12. *There is a* 13*-type classification scheme, such that* $q(S) \leq 6p - \phi(S)$*.*

In both proofs the classification scheme allows us to identify the super-faces that benefit the most from the local optimality of $\mathcal{C}$ and separate them from those that do less. For some good cases, we can obtain a much better gain than for others, e.g., in one of our classification types, $gain(f)$ is as high as $\frac{9}{2}$. In the bad cases, we will have to use the lower bound of $\frac{3}{2}$ for the gain, that holds in general for any super-face.

8.2.1 Proof of Proposition 8.5

Observation 8.3 immediately leads to a simple lemma which will prove helpful for the proof of Proposition 8.5.

Lemma 8.13. *Let* $e := uv$ *be a type-*2 *edge in* $G[S]$*, then the cross triangles* t_1 *and* t_2 *supported by* e *can not have the same landing component.*

Proof. Since both u and v are in S, there exists a path P from u to v in $G[S]$ containing only cactus edges and vertices. Hence, the cycle $D := uPv \cup vu$ consists of only type-1, type-2 or cactus edges and cactus vertices, such that the two cross triangles supported by e will be embedded in different regions corresponding to D. Thus, by Observation 8.3 the two cross triangles supported by e cannot have the same landing component. □

The 1-swap operation illustrated in Figure 8.1(a) and the 2-swap optimality of $\mathcal{C}$ imply the following lemma.

Lemma 8.14. *Let* t *be a cactus triangle with vertices* u, v *and* w *and let there exist at least two cross triangles* t_1 *and* t_2 *in* G *such that* $(V(t_1) \cup V(t_2)) \cap S_x^t \neq \emptyset$*, for* $x \in \{u, v, w\}$*, then*

(1) t_1 *and* t_2 *must have the same landing component,*

(2) any edge e *in* $B_{uv}^t \cup B_{uw}^t \cup B_{vw}^t$ *is of type-*1*,*

(3) $|B_{uv}^t|, |B_{uw}^t|, |B_{vw}^t| \leq 1$ *and*

(4) any set of edges $\{xy\} \cup B_{xy}^t$ *for* $xy \in E(t)$ *support at most one cross triangle.*

Proof. To prove Property (1), assume for contradiction that t_1 and t_2 do not share the same landing component. In this case we can increase the number of triangles in $\mathcal{C}$ by removing t from $\mathcal{C}$ and adding t_1 and t_2 to $\mathcal{C}$ in its place. As the landing components are disjoint this operation does not introduce any new cycle to $\mathcal{C}$ other than the supported cross triangles, and therefore the resulting structure is a cactus subgraph of G. This contradicts that $\mathcal{C}$ is 2-swap optimal.

Property (2) follows from Property (1). Assume for contradiction that there exists a type-2 edge $e \in B^t_{uw}$ (the same argument will hold for B^t_{uv} and B^t_{vw}). Only one of t_1 and t_2 can have its two cactus vertices in the same split-components of t as the endvertices of e. We may assume that this is not the case for t_1. Let t' and t'' denote the cross triangles supported by e. By Property (1) t' and t_1 must have the same landing component, the same holds for t'' and t_1. But by Lemma 8.13 t' and t'' can not have the same landing component, thus we reach a contradiction.

We will prove Property (3) also by contradiction. Assume that $|B^t_{vw}| \geq 2$ and let $e_1, e_2 \in B^t_{vw}$ be any two type-1 edges (the same argument will hold for B^t_{uv} and B^t_{uw}). As both endvertices of e_1 are cactus vertices, there exists a path in $\mathcal{C}[S]$ connecting both the endvertices, thus there is a cycle C_1 in $G[S]$ containing e_1 and only cactus edges otherwise. Similarly, there exists a cycle C_2 in $G[S]$ that contains e_2 and only cactus edges otherwise. In G either e_1 is embedded in the inside of the closed region bounded by C_2 or e_2 is embedded in the inside of the closed region bounded by C_1 (see Figure 8.5). For this proof we assume the former case. The proof for the latter case is symmetric.

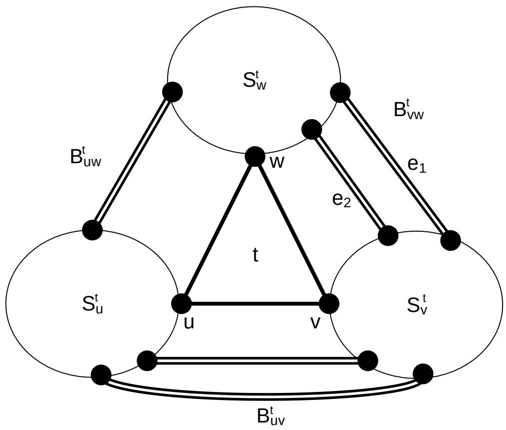


Figure 8.5: The split-components S^t_u, S^t_v, S^t_w and the sets $B^t_{uv}, B^t_{uw}, B^t_{vw}$ for a cactus triangle t. By Lemma 8.14 Property (3), the edges e_1 and e_2 cannot exist in a 2-swap optimal cactus subgraph.

Only one of t_1 and t_2 can have its two cactus vertices in the same split-components of t as the endvertices of e_1. We may assume that this is not the case for t_1. By Property (1) the cross triangle supported by e_1 and t_1 must have the same landing component. Note t_1 can not lie in the inside of the region bounded by C_1 in G. Therefore, the landing component shared by the two cross triangles must lie on the outside of C_1. However, by Property (1) the cross triangle supported by e_2 and t_1 must have the same landing component. We reach a contradiction using Observation 8.3.

We prove Property (4) also by contradiction. Assume that the set of edges $\{uv\} \cup B^t_{uv}$ supports two cross triangles (the same argument will hold for $\{uw\} \cup B^t_{uw}$ and $\{vw\} \cup B^t_{vw}$). Property (3) implies that there is only one type-1 edge in B^t_{uv} hence uv will support the other cross triangle. Let t' be the triangles supported by uv, t'' be the cross triangle supported by an edge $e' \in B^t_{uv}$. Only one of t_1 and t_2 can have its two cactus vertices in the same split-components of t as the endvertices of e. We may assume that this is not the case for t_1. By Property (1), t' and t_1 must have the same landing component. But this is also true for t'' and t_1. In addition, there is a cycle C in H that contains e and a path P from u' to v' in $\mathcal{C}[S]$ (where $u'v' = e$) containing only cactus vertices and edges such that t' is embedded in its inside in G and S^t_w outside of it. As t_1 intersects S^t_w it must be embedded outside of C in G. But by Observation 8.3, t' and t_2 cannot have the same landing component and we reach a contradiction. □

Further we can show that the following holds if a cactus triangle t of $\mathcal{C}$ supports two cross triangles.

Lemma 8.15. *If t supports cross triangles t_1 and t_2, where u denotes their common cactus vertex, then B^t_{uv} and B^t_{uw} are both empty.*

Proof. By Lemma 8.14, Property (1), t_1 and t_2 must have the same landing component. Note that if t_1 and t_2 have a common landing vertex, then the claim is trivially true, as then u is incident to exactly three faces, namely t,t_1 and t_2 which by definition are all empty and thus B^t_{uv} and B^t_{uw} are empty in this case. Thus, we assume that $t_1 \cap t_2 = u$.

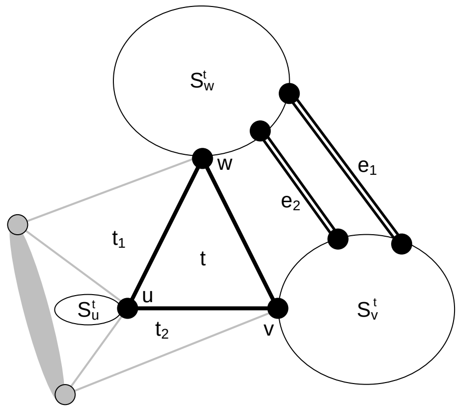


Figure 8.6: If t supports two cross triangles that intersect in a vertex $u \in S$, then by Lemma 8.15, B^t_{uv} and B^t_{uw} must both be empty.

Let u_1 and u_2 denote the landing vertices of t_1 and t_2 respectively. As t_1 and t_2 have the same landing component (say S'), G must contain a path P from u_1 to u_2 consisting of edges only in $\mathcal{C}[S']$. Furthermore $uu_1 \cup P \cup u_2u$ forms a cycle C with only one cactus vertex u and cross edges and edges in $\mathcal{C}[S']$. Note that the fact that t, t_1 and t_2 are empty in G, implies that the two cactus edges of t incident to u, as well as the edges uu_1 and uu_2 are consecutive in the circular edge incident list of u in G. This observation gives us two important facts. First, as C contains uu_1 and uu_2, any other edge incident to u in G must be embedded in the region bounded inside of C in G. Second, any split-components S^t_x, for $x \in \{v, w\}$, must be embedded outside of C in G. Assume for contradiction that there exists an edge e with endvertices u and $z \in S^t_x$, with $x \in \{v, w\}$, by the previous observation e has to cross C in G, and therefore the existence of e contradicts that G is a plane graph. Similarly, there cannot exist any edge e with one endvertices in $S^t_u \setminus \{u\}$ and another endvertices in S^t_x, with $x \in \{v, w\}$ since all these vertices are embedded *strictly* inside of C and S^t_x's, with $x \in \{v, w\}$, are embedded strictly outside of C. □

We are now ready to prove the different properties of heavy triangles claimed in Proposition 8.5. In the following, we will prove one lemma for every such claim.

Lemma 8.16. *Any cactus type-3 triangle t in $G[S]$ is light.*

Proof. For any vertex v in t, there is a pair of cross triangles supported by t such that their intersection is v, thus by Lemma 8.15, $B^t_{vv'}$ must be empty for any $v' \in V(t) \setminus v$. Hence, the number of cross triangles supported by $E(t)$ and $\cup_{ww' \in E(t)} B^t_{ww'}$ is less than four and each edge $vv' \in E(t)$ supports one cross triangle, thus t is a light triangle. □

Lemma 8.17. *Any cactus type-2 triangle t in $G[S]$ is light.*

Proof. Let t be a type-2 triangle, such that each of the cactus edges uw and vw support cross triangles t_1 and t_2 respectively (see Figure 8.7). By Lemma 8.15, B^t_{uw} and B^t_{vw} must be empty. By Lemma 8.14 properties (2) and (3) there is at most one edge in B^t_{uv} and if it exists it must be of type-1. Thus, there are at most three cross triangles supported by t and the edge in B^t_{uv} and in addition at least two edges in $E(t)$ support a cross triangle, thus t is a light triangle. □

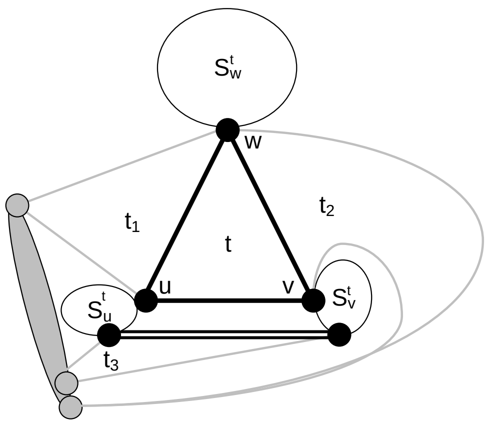


Figure 8.7: A type-2 light triangle t and an illustration of the third property of Proposition 8.21.

Lemma 8.18. *If t is a heavy type-1 triangle, with $V(t) = \{u, v, w\}$, let uv denote the edge in $E(t)$ that supports the cross triangle supported by t, then $B^t_{ww'} = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$ and the total number of cross triangles supported by edges in B^t_{uv} is greater than or equal to two.*

Proof. We first show that $B^t_{ww'}$ is empty for every $ww' \in E(t) \setminus uv$. Let t' denote the cross triangle supported by t. Assume for contradiction, that there exists an edge e in some $B^t_{ww'}$ for some edge $ww' \in E(t) \setminus uv$. As t' and the cross triangle supported by e fulfill the requirements of Lemma 8.14, Property (4) implies that there are at most three cross triangles supported by edges in $E(t) \cup_{vv' \in E(t)} B^t_{vv'}$, which contradicts the definition of a heavy triangle.

As B^t_{uw} and B^t_{vw} are empty there must be at least two cross triangles in G supported by edges in B^t_{uv}, as otherwise t would be light. □

Lemma 8.19. *If t is a heavy type-0 triangle, then there is an edge $uv \in E(t)$ such that $B^t_{ww'} = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$ and the total number of cross triangles supported by edges in B^t_{uv} is greater than or equal to three.*

Proof. We will first show that at most one of $B^t_{uu'}$ for $uu' \in E(t)$ can be non-empty. Assume for contradictions that there are two sets B^t_{uv} and B^t_{uw} which are non-empty. Then the cross triangles supported by the edges in these two sets fulfill the requirements of Lemma 8.14. Hence, $|B^t_{uv}|, |B^t_{uw}|, |B^t_{vw}| \leq 1$ and the number of cross triangles supported by $E(t) \cup B^t_{uv} \cup B^t_{uw} \cup B^t_{vw}$ is at most three, contradicting the fact that t is heavy.

Therefore, we know that there is only one edge $uv \in E(t)$ such that B^t_{uv} is non-empty. As t is heavy B^t_{uv} must contain edges that support at least three cross triangles as otherwise t would be light. □

8.2.2 Analyzing the Outer-Face f_0 (Proof of Lemma 8.9)

In this subsection, we will prove that $survive(f_0) \leq \mu(f_0) - \phi(S) + 1$. From this we easily follow that $\phi(S) - 1 \leq \mu(f_0) - survive(f_0) = gain(f_0)$. If $\phi(S) \leq 3$, this bound can easily be achieved by enumerating all possible compositions of the face boundary of f_0. If $\phi(S) > 3$, the $\phi(S)$ term in the bound we want to prove becomes more significant and hence this case needs special treatment.

In contrast to the other super-faces in $\mathcal{F}$, the number of surviving triangles in f_0 also depends on $\phi(S)$. We first give an intuition on how this term influences the number of surviving triangles in f_0 and then use the idea behind it to prove Lemma 8.9. Starting from $G[S]$, we can construct an auxiliary graph $\widetilde{G}$ by modifying the outer-face f_S, such that this part of the graph is fully triangulated using type-0 edges, such that in total we obtain $\phi(S) - 2$ extra triangles. Also, in this process the structure of the free and occupied edges of the outer-face (say $\widetilde{f_0}$) of the subgraph $\widetilde{H} := \widetilde{G} \setminus A$ (where A is the set of type-0 edges) of $\widetilde{G}$ remains exactly the same as that of the original outer-face f_0 of H. Finally, we use the trivial upper bound given by Lemma 8.38 on the number of triangular faces embedded inside the outer-face $\widetilde{f_0}$ in graph $\widetilde{G}$, which in turn gives us the $-\phi(S)$ term for the bound on the number of triangular faces embedded inside the outer-face f_0 in graph $G[S]$. Notice that the modified graph $\widetilde{G}$ is created only for counting purposes and the modification does not change the structure of our original graph G in any way. The following lemma formalizes this idea of triangulating the outer-face.

Lemma 8.20. *For the graph $G[S]$ with outer-face f_S having $\phi(S) > 3$ free edges, there exists another simple planar graph $\widetilde{G}$ with outer-face $\widetilde{f_S}$, such that*

- *The graphs $\widetilde{G}$ and G only differ inside the outer-face f_S of $G[S]$.*

- *The structure of the outer-face* $\widetilde{f}_0$ *of the graph* $\widetilde{H} := \widetilde{G} \setminus A$ *(where A is the set of type-*0 *edges) is the same as that of* f_0*, i.e.,* $|Occ(f_0)| = |Occ(\widetilde{f}_0)|$ *and* $|Free(f_0)| = |Free(\widetilde{f}_0)|$.
- *There are at least* $\phi(S) - 2$ *extra surviving triangles embedded inside the outer-face* $\widetilde{f}_0$ *in* $\widetilde{G}$ *as compared to the outer-face* f_0 *in* $G[S]$.

Proof. To prove this lemma, we will transform $G[S]$ to $\widetilde{G}$ by creating at least $\phi(S) - 2$ new surviving triangles in f_S by first pre-processing and then triangulating f_S using extra type-0 edges in a specific way.

First we *decouple* the supported cross triangles embedded inside f_S which share their landing components by adding a dummy landing vertex for each such cross triangle and making the new dummy vertex its landing component. Notice that the decoupling step makes the induced graph $G[V(f_S)]$ an outer-planar graph, where $V(f_S)$ are the vertices contained in face f_S. Also, it does not change the structure of the graph G anywhere else except inside face f_S. Since $G[V(f_S)]$ is outer-planar, there exists a vertex $u_1 \in V(f_S)$, such that the degree of u_1 in $G[V(f_S)]$ is two. Now we number the vertices in the face f_S in clockwise order as $u_1, u_2, \ldots u_{\ell_S}$, where u_1 is the degree two vertex in $G[V(f_S)]$. Next, we triangulate the outer-face f_S by adding a star of type-0 edges with vertex u_1 as the root of it and vertices $u_3, u_4 \ldots u_{\ell_S - 1}$ as the leaves of the star (see Figure 8.8). This completes the construction of our auxiliary graph $\widetilde{G}$. Notice that this operation cannot create a parallel edge in $\widetilde{G}$, implied by the way we fixed u_1. Also, the decoupling and triangulation will maintain the planarity of $\widetilde{G}$. Finally, it is easy to see that the occupied and the free edges of the outer-face $\widetilde{f}_0$ of graph $\widetilde{H}$ are the same as that of the original outer-face f_0, hence the second property is satisfied.

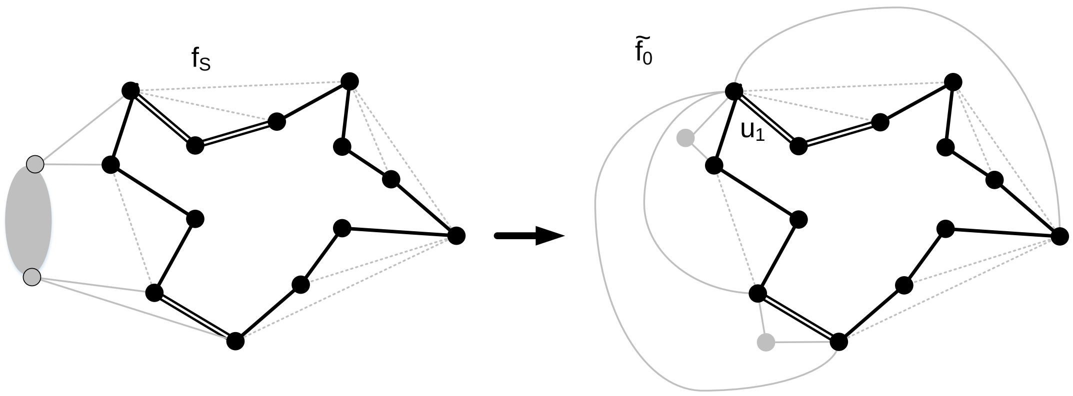


Figure 8.8: The decoupling and triangulation of the face f_S. On the left f_S is identical to the outer face of the drawn graph after deleting all cross edges and the landing component. On the right $\widetilde{f}_0$ can be formed by deleting all type-0 and cross edges.

Each of the triangles (u_1, u_2, u_3) and $(u_{\ell_S - 1}, u_{\ell_S}, u_1)$ could either survive if both the edges coming from f_S are free or not survive if at least one of these edges is occupied. Any triangle of the form (u_1, u_i, u_{i+1}) for $2 < i < \ell_S - 1$ will survive if the (u_i, u_{i+1}) edge is free. Now if both the triangles (u_1, u_2, u_3) and $(u_{\ell_S - 1}, u_{\ell_S}, u_1)$ do not survive, then at most two out of the $\phi(S)$ free edges can be a part of these triangles and hence there will be at least $\phi(S)-2$ triangles of the form (u_1, u_i, u_{i+1}) for $2 < i < \ell_S - 1$ which survive. If one of the triangles (u_1, u_2, u_3) and $(u_{\ell_S - 1}, u_{\ell_S}, u_1)$ survives, then at most three out of the $\phi(S)$ free edges can be part of these triangles and hence there will be at least $\phi(S)-3$ triangles of the form (u_1, u_i, u_{i+1}) for $2 < i < \ell_S - 1$ which survive. Else both of the (u_1, u_2, u_3) and $(u_{\ell_S - 1}, u_{\ell_S}, u_1)$ triangles survive, then four out of the $\phi(S)$ free edges will be part of these triangles and hence there will be at least $\phi(S)-4$ triangles of the form (u_1, u_i, u_{i+1}) for $2 < i < \ell_S - 1$ which survive. Hence, overall in each case, $\phi(S)-2$ triangles survive and the lemma follows. □

Note that, $|E(f_0)| \geq \ell_S \geq \phi(S)$ since f_S is formed after including all the $a_0(f)$ edges embedded inside f_0 in G (see Figure 8.9).

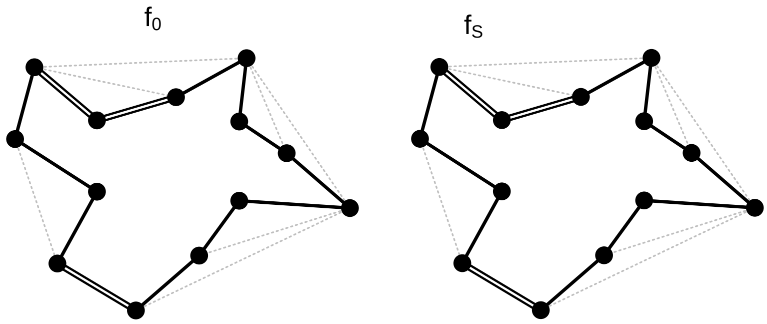


Figure 8.9: On the left, the outer face boundary resulting from deleting all type-0 edges corresponds to the outer super-face f_0 of H. On the right, the outer face corresponds to the outer face f_S of $G[S]$.

Now, we are ready to present the proof of Lemma 8.9. We split the analysis into two cases:

- First, consider the case when $|E(f_0)| = 3$. The worst case then is when $\phi(S) = 3$, which implies $|Free(f_0)| = 3$, $|Occ(f_0)| = 0$ and $\mu(f_0) = 3$. In this case, $survive(f_0) = 1$, which gives the inequality.

  Otherwise, when $\phi(S) \leq 2$, we have $survive(f_0) = 0$ (there would be an occupied edge that supports a cross triangle in f_0 which kills it), $|Free(f_0)| \leq 2$ and $|Occ(f_0)| \geq 1$. This gives $\mu(f_0) \geq \frac{3}{2}$, and $\mu(f_0) - \phi(S) + 1 \geq \frac{1}{2} > survive(f_0)$.

- If $|E(f_0)| > 3$ and $\phi(S) \leq 3$, then the trivial bounds given by Lemma 8.38 and 8.39 imply the inequality.

  From now on we assume that $\phi(S) > 3$. For this case, we use Lemma 8.20 on $G[S]$ to get the auxiliary graph $\widetilde{G}$ with at least $\phi(S)-2$ extra surviving faces in its outer-face, totaling to $survive(f_0) + \phi(S) - 2$. Now using the trivial bound given by Lemma 8.38 on the outer-face $\widetilde{f_0}$ for the corresponding graph $\widetilde{H}$, we get

  $$survive(f_0) + \phi(S) - 2 \leq survive(\widetilde{f_0}) \leq \mu(\widetilde{f_0}) - 2 \leq \mu(f_0) - 2,$$

  which concludes the proof of Lemma 8.9.

8.3 Reduction to Heavy Cacti

In this section, we will prove Theorem 8.4. For this, we assume that the bound $q(S) \leq \gamma p - \phi(S)$ holds for some $\gamma \geq 6$ and all S where the cactus subgraph $\mathcal{C}[S]$ contains only heavy triangles. We will show that this bound then also holds for any S that contains an arbitrary number of light triangles. We will prove this by induction on the number of light triangles in $\mathcal{C}[S]$. The proof does not require us to use the skeleton-graph H of G, we will, however, reuse some of the terminology introduced in the previous section. The base case (when all triangles are heavy) follows from the precondition and the trivial base case when $|S| = 1$ is clearly true. Now assume that $\mathcal{C}[S]$ contains at least one light triangle t. Our plan is to apply the induction hypothesis on the subgraphs $\{G[S_v^t]\}_{v \in V(t)}$ since each $\mathcal{C}[S_v^t]$ contains less light triangles than $\mathcal{C}[S]$.

Since we now deal with a cactus subgraph that does not only consist of heavy triangles, we first show the following proposition (whose proof will be presented in Subsection 8.3.2) about important structural properties of G that come with light triangles.

Proposition 8.21 (Structure of light triangles)**.** *If t is a light triangle in $\mathcal{C}[S]$, then the following statements hold:*

- *If t is a light type-0 triangle and $uv \in E(t)$, such that $B_{ww'}^t = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$, then the total number of cross triangles supported by edges in B_{uv}^t is at most two.*
- *If t is a light type-1 triangle and the edge $uv \in E(t)$ supports the cross triangle supported by t and $B_{ww'}^t = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$, then the total number of cross triangles supported by edges in B_{uv}^t is at most one.*
- *If t is a light triangle where edges in $\bigcup_{uv \in E(t)} B_{uv}^t \cup E(t)$ support either two or three cross triangles such that at least two different set of edges $\{uv\} \cup B_{uv}^t$ for $uv \in E[t]$ supports a cross triangle each, then each set of edges $\{uv\} \cup B_{uv}^t$ supports at most one cross triangle and all the supported cross triangles have the same landing component.*

Free and occupied edges: We call the edges in the outer-face f_S of $G[S]$ that contribute to $\phi(S)$ *free* (,i.e., the edges on the outer-face that do not support any cross triangle of G) and every other edge in f_S that is not free is called *occupied*. Let $o(S)$ be the total number of occupied edges in f_S. It follows that $\phi(S) = \ell(S) - o(S)$.

8.3.1 Inductive proof

We now show how to prove the induction step. Consider a light cactus triangle $t \in \mathcal{C}[S]$ with vertices $V(t) = \{u, v, w\}$. To upper bound $q(S)$, we break it further into two distinct terms $q' + q''$:

Definition of q' and q'': The term q' counts all triangles in $G[S]$ that have all three vertices in the same split-component of t, and the cross triangles in $G[S]$ that are supported by edges or triangles in $G[S_x^t]$ for some $x \in \{u, v, w\}$. As each split-components of t is also a cactus subgraph, by induction we have for $G[S_x^t]$ for all $x \in \{u, v, w\}$: $q(S_x^t) \leq \gamma p(S_x^t) - \phi(S_x^t)$. As q' is equal to the sum over $q(S_x^t)$ for all $x \in \{u, v, w\}$ we get

$$q' \leq \gamma(p-1) - (\phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t)) = \gamma p - (\phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t)) - \gamma.$$

The term q'' counts all remaining triangles in $q(S)$, i.e., the triangles whose vertices belong to at least two different split-components of t. We will proceed to show that

$$q'' \leq 6 + \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t) - \phi(S).$$

hence, upper bounding $q' + q''$ by the desired quantity for any $\gamma \geq 6$.

To this end, we again split q'' into two terms and upper bound their contributions separately. The first term, q_1'', is the number of cross triangles supported by the edges in $B_{uv}^t \cup B_{uw}^t \cup B_{vw}^t$ plus the cross triangles supported by t plus one for t itself. The second term, q_2'', is the number of "surviving" triangular faces in $G[S] \setminus (\bigcup_{x \in V(t)} G[S_x^t])$, that do not have any cross triangles of G embedded inside of it.

Note that by the definition of the light triangles, there are at most three cross triangles supported by the edges in $B_{uv}^t \cup B_{uv}^t \cup B_{vw}^t$ and t itself. Now we consider two cases of how these cross triangles can be composed, based on the value of q_1''.

- (There exist at most two cross triangles supported by the edges in $B_{uv}^t \cup B_{uv}^t \cup B_{vw}^t \cup E(t)$): In this case q_1'' can be at most three, i.e., t itself and the supported cross triangles. Hence, showing that $q_2'' \leq 3 + \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t) - \phi(S)$, would complete the proof of the induction step.

- (There exist exactly three cross triangles supported by the edges in $B_{uv}^t \cup B_{uv}^t \cup B_{vw}^t \cup E(t)$): In this case $q_1'' = 4$, i.e., we count t itself and the three supported cross triangles. Hence, showing that $q_2'' \leq 2 + \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t) - \phi(S)$ in this case, will complete the proof of the induction step.

The following lemma (which proof can be found in Subsection 8.3.3) covers both of these cases in the described way and therefore completes the proof of Theorem 8.4.

Lemma 8.22. *For any light triangle t, the number of surviving triangles q_2'' is at most $3 + \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t) - \phi(S)$. Moreover, if there are three cross triangles supported by the edges in $B_{uv}^t \cup B_{uw}^t \cup B_{vw}^t$ and t itself, then q_2'' is at most $2 + \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t) - \phi(S)$.*

8.3.2 Proof of Proposition 8.21

In this subsection, we will prove the properties stated in Proposition 8.21 about light triangles. Recall that for a light triangle the edges in $E(t) \cup_{uv \in E(t)} B_{uv}^t$ support at most three cross triangles.

Lemma 8.23. *If t is a light type-0 triangle with one edge $uv \in E(t)$ such that $B_{ww'}^t = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$, then the total number of cross triangles supported by edges in B_{uv}^t is at most two.*

Proof. This simply follows from the definition of heavy triangles. If there where more than two cross triangle supported by the edges in B_{uv}^t, then t would be a heavy triangle. □

Lemma 8.24. *If t is a light type-1 triangle where uv supports the cross triangle supported by t and $B^t_{ww'} = \emptyset$ for all $ww' \in E(t) \setminus \{uv\}$, then the total number of cross triangles supported by edges in B^t_{uv} is at most one.*

Proof. This simply follows from the definition of heavy triangles. If there was more than one cross triangle supported by the edges in B^t_{uv}, then t would be a heavy triangle. □

Lemma 8.25. *If t is a light triangle where the edges in $\bigcup_{uv \in E(t)} B^t_{uv} \cup E(t)$ support either two or three cross triangles such that at least two different sets of edges $\{uv\} \cup B^t_{uv}$ for $uv \in E[t]$ support a cross triangle each, then each set of edges $\{uv\} \cup B^t_{uv}$ supports at most one cross triangle and all the supported cross triangles have the same landing component.*

Proof. For any pair of cross triangles supported by edges in two different sets in $\{uv\} \cup B^t_{uv}$ for $uv \in E[t]$, Lemma 8.14 implies that both cross triangles must have the same landing component. Since there exists at least one pair of such triangles, by Lemma 8.14 Property (4), the claim of this lemma follows. □

8.3.3 Proof of Lemma 8.22

To facilitate the counting arguments that we will use to prove Lemma 8.22, we will be working with an auxiliary graph $\widetilde{G}$ instead of $G[S]$. Let Γ_x denote the face boundary (in particular, the set of edges on the facial walk) of the outer-face of $G[S^t_x]$ for $x \in \{u, v, w\}$ and let Γ denote the face boundary of the outer-face of $G[S]$ (so Γ contains exactly all the outer-edges). Because $\mathcal{C}[S]$ is a connected triangular cactus, there cannot be any repeated edge in these facial walks, hence Γ, Γ_i's are circuits; some vertices may occur multiple times in Γ_x or Γ. Now we *cut open* each of the circuits Γ,Γ_x, for each $x \in \{u, v, w\}$ to convert them to simple cycles. The idea is to make copies of each vertex contained in the circuit (the number of copies will be equal to the number of times it appears in the corresponding circuit) and joining the edges incident to the original vertex to one of the copies, such that important structures of the original embedding are preserved. We also make sure that there exists a triangular face corresponding to t containing some copy of each of the vertices in $\{u, v, w\}$. After we cut-opened, Γ_x, for each $x \in \{u, v, w\}$ will be an empty cycle in $\widetilde{G}$. Notice that the values of ϕ as well as the types of edges on these cut-opened cycles are preserved.

Note that the surviving triangles that contribute to q''_2 correspond exactly to the triangles embedded in the regions of G exterior of Γ_x for all $x \in \{u, v, w\}$ but in the interior of Γ. Also, t is embedded inside of Γ. In order to bound q''_2 we construct an auxiliary graph $\widetilde{G}$ as follows. For each $x \in \{u, v, w\}$, we remove all edges and vertices embedded in the interior of cycle Γ_i from $G[S]$. The resulting graph after such a removal is our $\widetilde{G}$, such that $V(\widetilde{G}) = V(\Gamma) \cup V(\Gamma_u \cup \Gamma_v \cup \Gamma_w) = V(\Gamma_u \cup \Gamma_v \cup \Gamma_w)$. Any triangle that contribute to the term q''_2 also exist as triangular faces in $\widetilde{G}$, thus we only need to upper bound $f_3(\widetilde{G})$.

Claim 8.26. *If $E(\Gamma) \setminus (E(t) \cup E(\Gamma_u \cup \Gamma_v \cup \Gamma_w)) = \emptyset$, then the bound for q'' holds.*

Proof. If the set is empty, then $q''_2 = 0$ and $\phi(S) \leq \phi(S^t_u) + \phi(S^t_v) + \phi(S^t_w) + 3$ in general. In the three cross triangles case, having no such edge implies that t is a type-

3 triangle, because all three cross triangles have to be supported by $E(t)$ and hence $\phi(S) = \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t)$. □

Now we continue with the case where there exists at least on edge in $E(\Gamma) \setminus (E(t) \cup E(\Gamma_u \cup \Gamma_v \cup \Gamma_w))$. Clearly, $\widetilde{G}$ is a subgraph of $G[S]$ and any surviving triangle in G must be embedded in a region of $\widetilde{G}$. To bound the number of surviving triangles corresponding to q_2'', we will first identify these regions and then make a region-wise analysis to get the full bound. For this purpose, we remove any non-cactus edge from $\widetilde{G}$ that is embedded in the interior of Γ and does not belong to one of Γ_u, Γ_v or Γ_w to form another auxiliary graph $\widetilde{G}'$. The faces in the graph $\widetilde{G}'$ which are embedded inside the cycle Γ and outside every cycle Γ_x (except the triangular face t), will correspond to the regions in $\widetilde{G}$ which we would analyze later. First, we state the following claim which quantifies the structure of these regions (see Figure 8.10 which illustrates all possible compositions of these regions). We will give the prof for the claim in a later subsection.

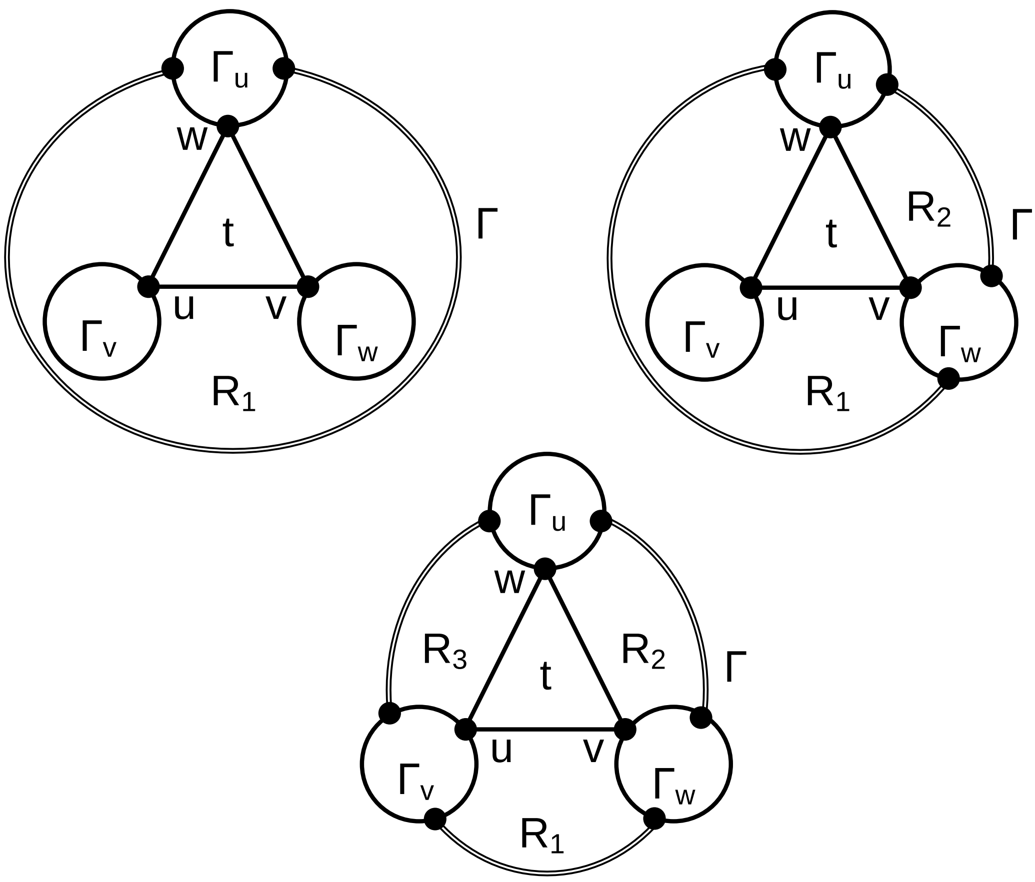


Figure 8.10: An illustration of the three possible shapes for $k \in \{1, 2, 3\}$ and the faces $R_1, \ldots R_k$ of $\widetilde{G}'$.

Claim 8.27. *If $R_1, \ldots, R_k$ (except the triangular face t) are the faces in $\widetilde{G}'$ which are embedded inside Γ and outside every cycle Γ_x for each $x \in \{u, v, w\}$, then $1 \leq k \leq 3$.*

Moreover, every such face contains exactly one edge of Γ*.*

Let $R_1, \ldots, R_k$ (for $1 \leq k \leq 3$) be the regions in $\widetilde{G}$ which are the faces of $\widetilde{G}'$ given by the above claim (see Figure 8.10 for an illustration). We denote by $\ell(R_i)$[1] the overall number of edges and by $o(R_i)$ the number of occupied edges in the boundary of R_i (these are the edges belonging to some cycle Γ_x for $x \in \{u, v, w\}$). In the next step, we will upper bound the number of surviving triangles that exist in G in each such region R_i.

Observation 8.28. *Any face in the graph* $\widetilde{G}$ *which is embedded inside one of the regions* R_i *contains vertices from at least two cycles* Γ_x, Γ_y *for* $x, y \in \{u, v, w\}$ *and* $x \neq y$*.*

How many surviving triangles can there be in region R_i? Intuitively, if we triangulate R_i by adding edges in its interior, we would have $\ell(R_i) - 2$ triangular faces. Among these faces, $o(R_i)$ of them would not be surviving since the edge bounding the face is occupied. In certain cases, we would get an advantage and the term would become -3 instead of -2.

Claim 8.29. *The number of surviving triangles embedded inside* R_i *in* $\widetilde{G}$ *are at most* $\ell(R_i) - o(R_i) - 2$*. Moreover, if the common landing component* L *of the three cross triangles supported by* $B^t_{uv} \cup B^t_{vw} \cup B^t_{uw} \cup E(t)$ *is embedded inside* R_i*, then we get the stronger bound of* $\ell(R_i) - o(R_i) - 3$*.*

The proof of this claim relies on a standard triangulation trick used in the context of planar graphs. We defer the proof to later in Subsection 8.3.5.

Now we are ready to complete the proof of Lemma 8.22. Let $\mathbf{1}^t_S \in \{0, 1\}$ be the indicator variable such that $\mathbf{1}^t_S = 1$ if we are in the case when there exists exactly three cross triangles supported by $B^t_{uv} \cup B^t_{vw} \cup B^t_{uw} \cup E(t)$ such that the common landing component L of these triangles is embedded inside some region R_i, otherwise $\mathbf{1}^t_S = 0$. Using the bounds for each region from Claim 8.29 we can upper bound q''_2 by summing over the number of surviving triangles in each region.

$$\begin{aligned} q''_2 &\leq \sum_{i=1}^{k} (\ell(R_i) - o(R_i) - 2) - \mathbf{1}^t_S \\ &\leq \sum_{i=1}^{k} \ell(R_i) - \sum_{i=1}^{k} o(R_i) - 2k - \mathbf{1}^t_S. \end{aligned} \tag{8.4}$$

Next, we take a closer look at the $\ell(R_i)$ term in the sum. By Claim 8.27, each region R_i contains exactly one edge of Γ, and $R_i \subseteq \Gamma \cup E(t) \cup \left(\bigcup_{x \in V(t)} \Gamma_x\right)$. Therefore, we can decompose the length of face R_i into three parts:

$$\ell(R_i) = 1 + \sum_{x \in V(t)} |E(R_i) \cap \Gamma_x| + |E(R_i) \cap E(t)|.$$

By plugging this into Equation (8.4) we get,

[1] Notice that we slightly abuse the notation $\ell(\cdot)$ here. Before, we use $\ell(S)$ where S is a subset of cactus vertices, and now we are using $\ell(R)$ where R is a cycle bounding a region.

$$q_2'' \leq \sum_{i=1}^{k}(1 + \sum_{x \in V(t)} |E(R_i) \cap \Gamma_x| + |E(R_i) \cap E(t)|]) - \sum_{i=1}^{k} o(R_i) - 2k - \mathbf{1}_S^t \tag{8.5}$$

$$\leq \sum_{i=1}^{k}(\sum_{x \in V(t)} |E(R_i) \cap \Gamma_x| + |E(R_i) \cap E(t)|) - \sum_{i=1}^{k} o(R_i) - k - \mathbf{1}_S^t. \tag{8.6}$$

Note that t can not contribute more than its three edges to the boundaries of all k regions, thus $\sum_{i=1}^{k} |E(R_i) \cap E(t)| \leq 3$. Using this in Equation (8.5), we get

$$q_2'' \leq 3 + \sum_{i=1}^{k} \sum_{x \in V(t)} |E(R_i) \cap \Gamma_x| - \sum_{i=1}^{k} o(R_i) - k - \mathbf{1}_S^t. \tag{8.7}$$

Claim 8.30. $\sum_{i=1}^{k} \sum_{x \in V(t)} |E(R_i) \cap \Gamma_x| = \ell(S_u^t) + \ell(S_v^t) + \ell(S_w^t) - \ell(S) + k$

Proof. Notice that the sum on the left-hand-side counts all edges in $(\bigcup_{x \in V(t)} \Gamma_x) \setminus \Gamma$ where each edge is counted exactly once, and this contribution is $\sum_{x \in V(t)} \ell(S_x^t) - \ell(S)$. Additionally, by Claim 8.27, each edge in $\Gamma \setminus (\bigcup_{x \in V(t)} \Gamma_x)$ is also counted exactly once as well, and this contribution is $+k$. □

Combining all of this with Inequality (8.7) we get,

$$q_2'' \leq 3 + \ell(S_u^t) + \ell(S_v^t) + \ell(S_w^t) - \ell(S) - \sum_{i=1}^{k} o(R_i) - \mathbf{1}_S^t. \tag{8.8}$$

.

Let $o_{across}^t(S)$ be the number of occupied edges among the $o(S)$ occupied edges belonging to Γ such that they do not belong to any of the Γ_x for $x \in \{u, v, w\}$. These edges are the ones which are embedded across two different cycles Γ_x, Γ_y for $x, y \in \{u, v, w\}$ and $x \neq y$ (potentially some of the edges embedded in double-line style in Figure 8.10). Hence, $o_{across}^t(S)$ captures precisely the number of occupied edges in $\Gamma \setminus (E(t) \cup \bigcup_{x \in V(t)} \Gamma_x)$ for which the supported cross triangles are embedded in the exterior of Γ. By the way we define $o(R_i)$, the following equality holds.

$$\sum_{i=1}^{k} o(R_i) = o(S_u^t) + o(S_v^t) + o(S_w^t) - (o(S) - o_{across}^t(S)). \tag{8.9}$$

Using this in Inequality (8.8) we get,

$$\begin{aligned} q_2'' &\leq 3 + \ell(S_u^t) + \ell(S_v^t) + \ell(S_w^t) - \ell(S) - (o(S_u^t) + o(S_v^t) + o(S_w^t) - (o(S) \\ &\quad - o_{across}^t(S))) - \mathbf{1}_S^t \\ &\leq 3 + (\ell(S_u^t) - o(S_u^t)) + (\ell(S_v^t) - o(S_v^t)) + (\ell(S_w^t) - o(S_w^t)) - (\ell(S) - o(S)) \\ &\quad - o_{across}^t(S) - \mathbf{1}_S^t. \end{aligned}$$

Since $\ell(S_x^t) = \phi(S_x^t) + o(S_x^t)$ for every $x \in \{u, v, w\}$, we get

$$q_2'' \leq 3 + \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t) - \phi(S) - o_{across}^t(S) - \mathbf{1}_S^t. \tag{8.10}$$

The general inequality $q_2'' \leq 3 + \phi(S_u^t) + \phi(S_v^t) + \phi(S_w^t) - \phi(S)$ for Lemma 8.22 trivially follows from the above inequality. The following claim will complete the proof.

Claim 8.31. *If there are three cross triangles supported by edges in $\bigcup_{uv \in E(t)} B_{uv}^t \cup E(t)$ with the common landing component L, then $o_{across}^t(S) + \mathbf{1}_S^t \geq 1$.*

Proof. There could be two sub-cases: (i) The landing component L is in the exterior of Γ. In this case, by the definition of $o_{across}^t(S) \geq 1$, all three edges which support one of the three cross triangles will contribute to $o_{across}^t(S)$ (see Figure 8.11 for illustration); and (ii) The cross triangles are embedded inside Γ. In this case, we have that $\mathbf{1}_S^t = 1$. In any case, we have $o_{across}^t(S) + \mathbf{1}_S^t \geq 1$, thus proving the lemma. □

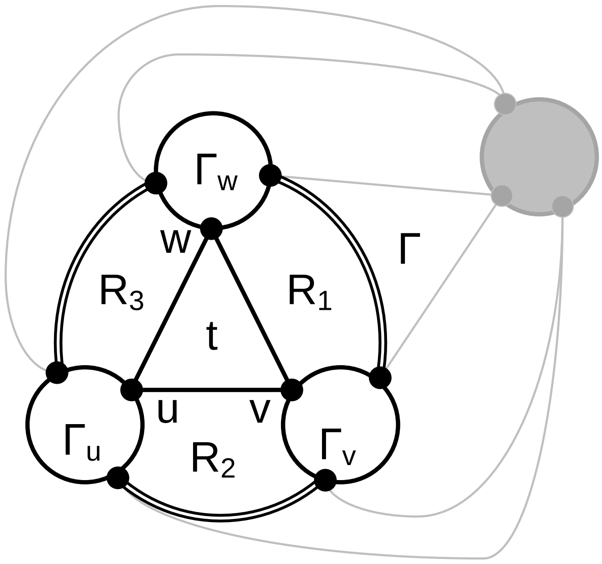


Figure 8.11: An example where three cross triangles are embedded in the exterior of Γ. This can only happen if there exist regions R_1, R_2 and R_3.

8.3.4 Proof of Claim 8.27

By the assumption that there exists at least one edge in $E(\Gamma) \setminus (E(t) \cup E(\Gamma_u \cup \Gamma_v \cup \Gamma_w))$. Let $ab := e \in E(\Gamma) \setminus (E(t) \cup E(\Gamma_u \cup \Gamma_v \cup \Gamma_w))$ be one such edge.

To prove the claim, we will show that for any such edge, there exists a unique face R satisfying the conditions of the claim and it contains at least one edge from $E(t)$. As each edge of t is also incident to the face bounded by t, this would imply that there can not be more than three such faces in $\widetilde{G}'$ and since there exists the edge e, hence we will be done.

Let $a \in \Gamma_x$ for some $x \in \{u, v, w\}$. We will always use the fact that since $e \in E(\Gamma)$, there are two directions starting from a to traverse the boundary of Γ_x, such that in one direction edges of Γ_x belongs to Γ and in the other they are embedded in the interior of Γ. Now we split into two possible cases.

- ($b \in \Gamma_y$ for some $y \in \{u, v, w\}$ such that $y \neq x$): Since $a, x \in \Gamma_x$, there exists a path P_x from a to x containing edges of Γ_x such that all these edges are embedded in the interior of Γ (possibly $x = a$ and P_x is a zero length path). Similarly there exist a path P_y going from b to y containing edges of Γ_y such that all these edges are embedded in the interior of Γ. Hence, the circuit C which includes the edge e, the edge $xy \in E(t)$ and two paths P_x and P_y, is embedded inside of Γ (except the edge e which is on the boundary of Γ). Clearly, there cannot be any other edge from Γ which is embedded inside C, hence any face embedded inside C can contain at most the edge e from $E(\Gamma) \setminus (E(t) \cup E(\Gamma_u \cup \Gamma_v \cup \Gamma_w))$. Also, by the way we define $\widetilde{G}'$, there cannot be any other edge inside C embedded across different Γ_i cycles. Now if t is embedded outside of C, then C itself is the face R of $\widetilde{G}'$ satisfying our requirements. Otherwise, the whole of Γ_z for $z \neq x$ and $z \neq y$, is embedded inside of C. This means that region inside the circuit C can be decomposed into the triangular face t, the cycle Γ_z and another face R whose boundary comprises of edges $xz, zy \in E(t)$, the edge of Γ_z, the edge e and two paths P_x and P_y. Hence, R is the face corresponding to e which we require.

- ($b \in \Gamma_x$): Notice that in this case, the circuit comprising of edge e along with a path P_x from a to b containing edges of Γ_x such that all these edges are embedded in the interior of Γ, will enclose the triangle t and the other two cycles Γ_y, Γ_z such that $y, z \in \{u, v, w\}$ and $x \neq y \neq z \neq x$. Similar to the previous case, there cannot be any other edge from Γ which is embedded inside C, C is embedded in the interior of Γ (except the edge e which is on the boundary of Γ) and also no other edge is embedded across different Γ_i cycles inside of C. Hence, any face embedded inside C can contain at most the edge e from $E(\Gamma) \setminus (E(t) \cup E(\Gamma_u \cup \Gamma_v \cup \Gamma_w))$. Also, C can be decomposed into the triangular face t, the cycles Γ_y, Γ_z and another face R whose boundary comprises of edge e, all three edges of t, all the edge of Γ_y, Γ_z and two paths P and P' from a to x and x to b containing edges of Γ_x embedded inside Γ. Hence, R is the face corresponding to e which we require.

8.3.5 Proof of Claim 8.29

To prove Claim 8.29 we will perform a series of monotone operations within the region R_i in graph $\widetilde{G}$, such that in each operation the number of surviving triangles embedded within R_i cannot reduce. In the end, we will reach a structure for which the bound holds trivially. Since the operations here are monotone, the bound which we get also holds for the original number of surviving triangles embedded within R_i. Notice that we make these modifications in the auxiliary graph $\widetilde{G}$ only for counting purposes and never change the structure of our graph G.

In the first step, except for the three cross triangles supported by the edges in $B^t_{uv} \cup B^t_{vw} \cup B^t_{uw} \cup E(t)$, we *decouple* all the other supported cross triangles embedded inside R_i which share their landing components by adding a dummy landing vertex for

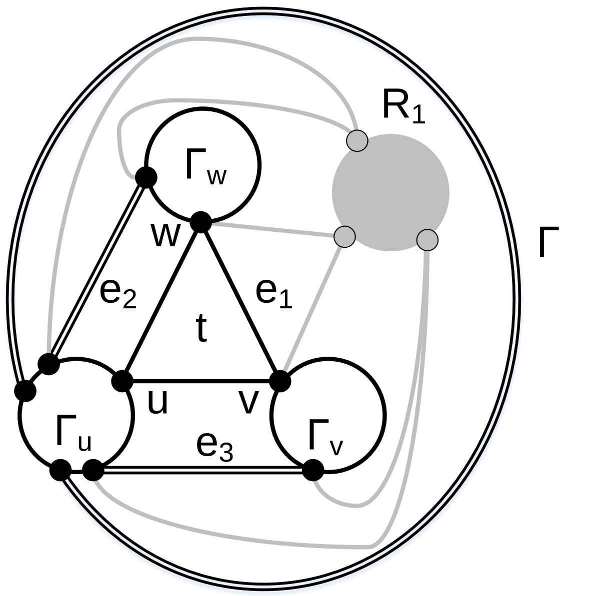


Figure 8.12: The case when there are three cross triangles embedded in the interior of Γ. This can only happen if there exists only region R_1.

each such cross triangles and making the new dummy vertex its landing component. Note that the decoupling step allows us to get a full triangulation of R_i in its interior (except the face containing the common landing component L) and at the same time does not affect the number of surviving triangles embedded inside R_i in $\widetilde{G}$.

After this we triangulate the interior of R_i by adding extra type-0 edges, such that the endvertex for each additional edge lies in two different Γ_x and Γ_y for $x \neq y$. This is possible to achieve due to Observation 8.28 and also this operation is monotone and cannot reduce the number of surviving triangles embedded inside R_i in $\widetilde{G}$. Also, all the faces inside R_i are triangular faces except the one containing L in graph $\widetilde{G}$. The way we triangulate the regions of R_i ensures that the Observation 8.28 continues to hold which implies that any face in R_i can contain at most one edge from the boundary of Γ_x for any $x \in \{u, v, w\}$. Also, $\widetilde{G}$ will remain a simple planar graph since the added type-0 edge connect vertices from the boundary of two different cycles Γ_x and Γ_y for $x \neq y$. In the end, we have at most $\ell(R_i) - 2$ triangular faces and any occupied edge counted in $o(R_i)$ (i.e., occupied edges in R_i which belongs to some cycle Γ_x for $x \in \{u, v, w\}$) can kill at most one triangle, hence the claimed upper bound follows in the general case.

Now in the case where we have the three cross triangles supported by the edges in $B^t_{uv} \cup B^t_{vw} \cup B^t_{uw} \cup E(t)$, we will prove that the face (say f) of R_i inside which the common landing component L is embedded, contains at least one more edge in addition to the three edges from $B^t_{uv} \cup B^t_{vw} \cup B^t_{uw} \cup E(t)$ which supports the three cross triangles. This implies that this face has length at least four and the triangulation of R_i misses at least two triangular faces. Also, in the worst case, the fourth edge which we consider here could contribute to the term $o(R_i)$. Hence, overall, we get at least 1 less surviving triangular face than the previous bound and the claim follows.

To prove the claim for face f, first recall that (by Proposition 8.21) the three edges in $B^t_{uv} \cup B^t_{vw} \cup B^t_{uw} \cup E(t)$ which supports the cross triangles are embedded across different pair of cycles $\Gamma_u, \Gamma_v, \Gamma_w$. Let $e_1 \in B^t_{vw} \cup vw$, $e_2 \in B^t_{uw} \cup uw$ and $e_3 \in B^t_{uv} \cup uv$ be the three edges supporting the three cross triangles. There is a cycle C comprising of edges e_1, e_2, e_3 and paths P_u, P_v, P_w joining the two ends of these edge in $\Gamma_u, \Gamma_v, \Gamma_w$ respectively, such that the triangle t is embedded inside C and the exterior of the Γ is outside of C. Now since R_i is a bounded region in graph $\widetilde{G}$ hence the face f is a bounded face. Now we show that for f to be a bounded face, its length has to be at least four. In the corner case when $e_1 = vw, e_2 = uw, e_3 = uv$, C is precisely the triangular face t and the edges are e_1, e_2, e_3 are touching f from the outside of t. Hence, for f to be bounded, there should exist at least one more edge to complete the loop going from Γ_u to Γ_v to Γ_w and back to to Γ_u. Otherwise, assume $e_1 \neq vw$ (other cases are symmetric). Since the cross triangles supported by e_1, e_2, e_3 share their landing component, and there exists a cycle C' containing only cactus/type-0/type-1/type-2 edges including edges e_1, vw and paths in Γ_v and Γ_w connecting the endvertices of e_1 and vw, such that the face f should be embedded outside of C. Now again for f to be bounded, it should contain one more edge and we are done.

8.4 A Classification Scheme for Factor Seven

In this section, we will present a classification scheme that allows us to prove Theorem 8.11. For simplicity, from now on we will use q instead of $q(S)$. More precisely, the aim is to prove the following lemma.

Lemma 8.32. *There is a* 5*-type classification scheme for which*

$$-(\sum_{f \in \mathcal{F}} gain(f)) \leq -\phi(S) + (2p + \frac{1}{2}p_1 - \frac{5}{2}a_1 - 3a_2 - \frac{3}{2}).$$

First, we show that Lemma 8.32 is sufficient for proving Theorem 8.11. For this, we substitute the bound from Lemma 8.32 into Inequality (8.2) to get:

$$q \leq (4p + \frac{1}{2}p_1 + \frac{5}{2}a_1 + 3a_2) - \phi(S) + (2p + \frac{1}{2}p_1 - \frac{5}{2}a_1 - 3a_2 - \frac{3}{2}) = 6p + p_1 - \phi(S) - \frac{3}{2}.$$

This implies $q \leq 7p - \phi(S)$ as desired. In order to define the classification schemes, we further classify the edges, vertices, and split-components for any heavy triangle t in $G[S]$ into several types.

Further classification of cactus vertices, edges and split-components: The cactus edges of each heavy triangle are further classified into *free* and *base* edges as follows: For any heavy triangle t, with vertices $V(t) = \{u, v, w\}$. Let $uv \in E(t)$ be an edge for which $B^t_{uv} \neq \emptyset$. By Proposition 8.5 there is exactly one such edge in $E(t)$. We say that the edge uv is the *base* edge and both u and v are called *base* vertices. We say that the other two edges in $E(t) \setminus uv$ are *free*, and the vertex w is called a *free* vertex. Both S^t_u and S^t_v are called *occupied* components and S^t_w is a *free* component. See Figure 8.13 for an illustration. The following claim follows from the properties of heavy type-0 and type-1 triangles shown in Proposition 8.5.

Claim 8.33. *The two free cactus edges of any cactus triangle are part of the same super-face in $\mathcal{F}$.*

Proof. Let vw and uw be the free edges in $E(t)$. Assume for contradiction that there is a super-face $f \in \mathcal{F}$ that only contains uw but not vw. Any super-face boundary needs to contain at least one type-1 or type-2 edge in order to form a cycle. Therefore, a path along the super-face f, not including the edge uw, from u to w must leave S^t_w using a type-1 or type-2 edge, a contradiction to the fact that for a heavy triangle, B^t_{vw} and B^t_{uw} are empty in graph H. □

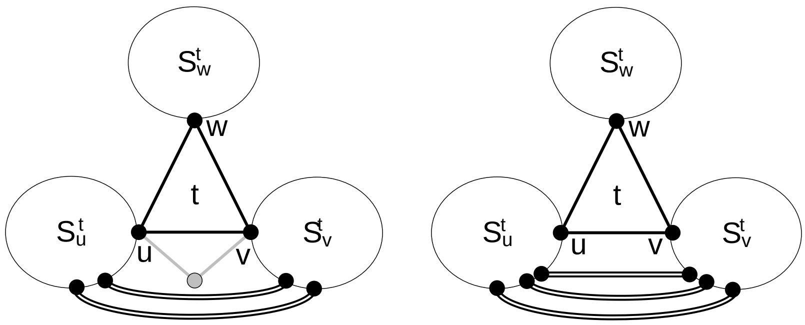


Figure 8.13: An illustration on how we classify cactus edges and split-components, based on a heavy triangle t and the type-1 and type-2 edges going across its split-components. The split-components S^t_u, S^t_v are occupied components and S^t_w is the free component of t. For a type-1 triangle (left figure), we know that the edge uv must also supports a cross triangle.

We will upper bound the number of surviving triangles inside any super-face $f \in \mathcal{F}$ based on the characteristics of the edges bounding f (see Figure 8.14).

Classification of Edges in the Face Boundaries of H: Edges that bound f are further partitioned into the following types:

- The two free edges of each cactus triangle. Let $p_0^{free}(f)$ and $p_1^{free}(f)$ denote the total number of type-0 and type-1 triangles respectively whose free edges participate in f.
- The base edges of the cactus triangles. Let $p_0^{base}(f)$ and $p_1^{base}(f)$ denote the total number of such triangles whose base edges participate in f.
- The type-2 edges. Let $a_2(f)$ denote the total number of such edges on f.
- The type-1 edges whose supported cross triangle are embedded inside f. This side of any type-1 edge is referred to as the *occupied* side. Let $a_1^{occ}(f)$ denote the total number of such edges in the boundary of f.
- The type-1 edges whose supported cross triangles are embedded in G in some region bounded by a super-face other than f. This side of any type-1 edge which

does not support a cross triangle is referred to as the *free* side. We denote the number of such edges by $a_1^{free}(f)$.

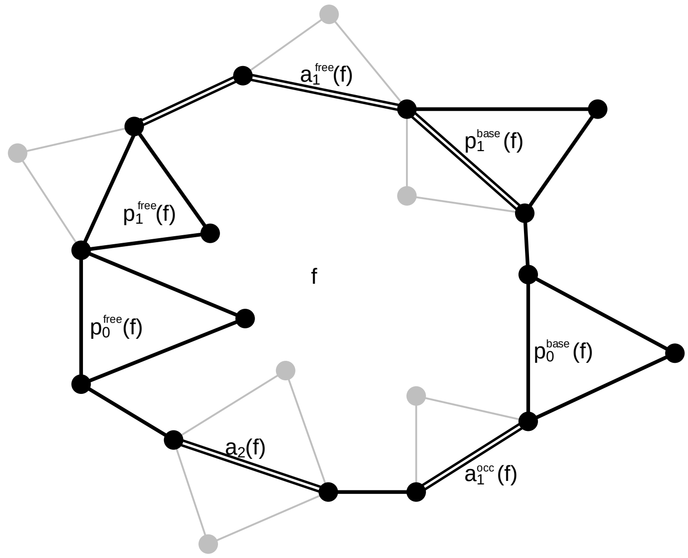


Figure 8.14: An example for the different types of edges in a super-face $f \in \mathcal{F}$ (to see the actual region, one has to ignore all cross edges in this graph). For each type we indicate to which quantity they contribute.

Notice that $|\mathcal{F}| > 1$ since all cactus triangles in $G[S]$ are heavy, hence $a_1 + a_2 \geq 1$. Since $\mathcal{C}$ is a triangular cactus and $|\mathcal{F}| > 1$, the following can be observed.

Observation 8.34. *For any super-face $f \in \mathcal{F}$, $a_2(f) + a_1(f) \geq 1$.*

Let $p^{free}(f) := p_0^{free}(f) + p_1^{free}(f)$, $p^{base}(f) := p_0^{base}(f) + p_1^{base}(f)$ and $a_1(f) := a_1^{occ}(f) + a_1^{free}(f)$.

Observation 8.35. *Any surviving triangular face cannot be incident to any type-2 edge, the occupied side of a type-1 edge or the base side of a type-1 triangle.*

By Observation 8.35 and 8.34, $|E(f)| = 2p^{free}(f) + p^{base}(f) + a_2(f) + a_1(f)$. Also, $|Free(f)| = 2p^{free}(f) + a_1^{free}(f) + p_0^{base}(f)$ and $|Occ(f)| = a_1^{occ}(f) + a_2(f) + p_1^{base}(f)$.

8.4.1 Classification Rules

Now we are ready to define the classification rules for our analysis. Since the bound on the number of surviving triangles (hence the $gain(f)$ quantity) that can be embedded inside each super-face heavily depends on the type of edges contained in its face boundary, we classify each super-face $f \in \mathcal{F}$ (except the outer-face f_0) into three broad categories, based on the total number of edges that contribute to $p_1^{base}(f) + a_2(f) + a_1(f)$. We also sub-categorize each super-face $f \in \mathcal{F}$ for which $p_1^{base}(f) + a_2(f) + a_1(f) = 1$ into further classes, based on whether it contains an $a_1^{free}(f)$ edge or not.

Classifications of super-faces: A super-face f will be of type-$[i,j]$ if $p_1^{base}(f)+a_2(f)+a_1(f)=i$ and $a_1^{free}(f)=j$. If there is no restriction on some dimension, then we put a dot ($[\bullet]$) there. Following is the precise categorization for the super-faces in $\mathcal{F}\setminus\{f_0\}$.

- A super-face f is of type-$[1,\bullet]$), if $p_1^{base}(f)+a_2(f)+a_1(f)=1$. In addition,
  - f is of type-$[1,0]$, if $a_1^{free}(f)=0$ or
  - of type-$[1,1]$), if $a_1^{free}(f)=1$.
- A super-face f is of type-$[2,\bullet]$, if $p_1^{base}(f)+a_2(f)+a_1(f)=2$
- A super-face f is of type-$[\geq 3,\bullet]$, if $p_1^{free}(f)+a_2(f)+a_1(f)\geq 3$

Let the set $\mathcal{F}[i,j]\subseteq\mathcal{F}$ be the subset of type-$[i,j]$ super-faces in H and analogously let $\eta[i,j]=|\mathcal{F}[i,j]|$ for each type-$[i,j]$ super-face. Notice that $\mathcal{F}[1,\bullet]\cup\mathcal{F}[2,\bullet]\cup\mathcal{F}[\geq 3,\bullet]\cup\{f_0\}=\mathcal{F}$ and $\mathcal{F}[i,\bullet]\cap\mathcal{F}[j,\bullet]$ for any $i\neq j$, which implies, $|\mathcal{F}|=1+\eta[1,\bullet]+\eta[2,\bullet]+\eta[\geq 3,\bullet]$. Also, $\mathcal{F}[1,j]\subseteq\mathcal{F}[1,\bullet]$ for any $j\in\{0,1\}$, hence, $\eta[1,\bullet]=\eta[1,0]+\eta[1,1]$.

The following lemma (whose proof will appear in Subsection 8.4.3) gives lower bounds on the quantity $gain(f)$ for each type of super-face in $\mathcal{F}\setminus f_0$. For f_0 we will use Lemma 8.9.

Lemma 8.36. *For any super-face $f\in\mathcal{F}$, the following holds:*

(1) If f is of type-$[1,0]$), then $gain(f)\geq\frac{5}{2}$.

(2) If f is of type-$[1,1]$), then $gain(f)\geq 2$.

(3) If f is of type-$[2,\bullet]$), then $gain(f)\geq 2$.

(4) If f is of type-$[\geq 3,\bullet]$), then $gain(f)\geq\frac{3}{2}$.

Notice that the bounds in Lemma 8.36 for the gain of super-faces of type-$[1,0]$, type-$[1,1]$ and type-$[2,\bullet]$ are better than the trivial bound of $\frac{3}{2}$, which leads to the improvement from $\frac{15}{2}$ to seven.

8.4.2 Proof for Lemma 8.32

We apply Lemma 8.9 and Lemma 8.36 to $\sum_{f\in\mathcal{F}} gain(f)$, depending on the type of each super-face: In particular, this includes the lower bounds for each super-face of type-$[1,0]$, type-$[1,1]$, type-$[2,\bullet]$, type-$[\geq 3]$ and the outer-face f_0.

$$
\begin{aligned}
-(\sum_f gain(f)) &\leq (1-\phi(S)) - \sum_{f\in\mathcal{F}[1,0]}\frac{5}{2} - \sum_{f\in\mathcal{F}[1,1]} 2 - \sum_{f\in\mathcal{F}[2,\bullet]} 2 - \sum_{f\in\mathcal{F}[\geq 3,\bullet]}\frac{3}{2}\\
&= 1-\phi(S)-\frac{5}{2}\eta[1,0]-2\eta[1,1]-2\eta[2,\bullet]-\frac{3}{2}\eta[\geq 3,\bullet].
\end{aligned}
$$

Here we use the fact that $|\mathcal{F}|=\eta[1,\bullet]+\eta[2,\bullet]+\eta[3,\bullet]+1$.

$$
\begin{aligned}
-(\sum_f gain(f)) &\leq 1-\phi(S)-\frac{5}{2}(|\mathcal{F}|-1)+\boxed{\tfrac{1}{2}\eta[1,1]+\tfrac{1}{2}\eta[2,\bullet]+\eta[\geq 3,\bullet]}\\
&= 3.5-\phi(S)-\frac{5}{2}|\mathcal{F}|+\boxed{\tfrac{1}{2}\eta[1,1]+\tfrac{1}{2}\eta[2,\bullet]+\eta[\geq 3,\bullet]}. \qquad (8.11)
\end{aligned}
$$

Next, we deal with the "residual terms" highlighted in the formula above by the box. For this purpose, we present various upper bounds on the number of super-faces of a certain type:

Lemma 8.37 (Two upper bounds on the number of super-faces)**.** *The following upper bounds hold:*

(1) $\eta[1,1] \leq a_1$.

(2) $\eta[2,\bullet] + 2\eta[\geq 3,\bullet] \leq p_1 + |\mathcal{F}| - 2$.

Proof. We start by proving the first upper bound. Since $a_1^{free}(f) = 1$ for a type-$[1,1]$ super-face f and each type-1 edge can contribute to $a_1^{free}(f)$ to exactly one super-face in $\mathcal{F}$, we have that $\eta[1,1] \leq a_1$.

The second upper bound can be proven by a simple charging argument. To each super-face $f \in \mathcal{F}$, we give one unit of money to a certain set of edges on the super-face. In particular, each of the following types of edges gets a unit: (i) base of the type-1 cactus triangle, (ii) type-1 edge, and (iii) type-2 edge. Therefore, the total amount of money put into the system is exactly:

$$\sum_{f\in\mathcal{F}} (p_1^{base}(f) + a_1(f) + a_2(f)) = p_1 + 2a_1 + 2a_2 = p_1 + 2|\mathcal{F}| - 2.$$

Counting from a different viewpoint, each super-face of type-$[j,\bullet]$ receives at least j units of money, thus the total amount is at least $1 + \eta[1,\bullet] + 2\eta[2,\bullet] + 3\eta[\geq 3,\bullet] = |\mathcal{F}| + \eta[2,\bullet] + 2\eta[\geq 3,\bullet]$. This immediately implies the inequality:

$$|\mathcal{F}| + \eta[2,\bullet] + 2\eta[\geq 3,\bullet] \leq p_1 + 2|\mathcal{F}| - 2.$$

□

After applying Lemma 8.37 to Inequality (8.11), we get that

$$\begin{aligned} -(\sum_f gain(f)) &\leq 3.5 - \phi(S) - \frac{5}{2}|\mathcal{F}| + \frac{1}{2}(a_1 + p_1 + |\mathcal{F}| - 2) \\ &= \frac{5}{2} - \phi(S) - \boxed{2|\mathcal{F}|} + \frac{1}{2}a_1 + \frac{1}{2}p_1. \end{aligned} \tag{8.12}$$

Using equality $|\mathcal{F}| = a_1 + a_2 + 1$ in Inequality (8.12), we have:

$$\begin{aligned} -(\sum_f gain(f)) &\leq \frac{5}{2} - \phi(S) - 2a_2 - \frac{3}{2}a_1 - 2 + \frac{1}{2}p_1 \\ &= \frac{1}{2} - \phi(S) + \boxed{a_1 + a_2} - \frac{5}{2}a_1 - 3a_2 + \frac{1}{2}p_1. \end{aligned} \tag{8.13}$$

And finally by using Lemma 8.10 in Inequality (8.13) we reach:

$$-(\sum_f gain(f)) \leq \frac{1}{2} - \phi(S) + 2p - 2 - \frac{5}{2}a_1 - 3a_2 + \frac{1}{2}p_1 = -\phi(S) + (2p + \frac{1}{2}p_1 - \frac{2}{5}a_1 - 3a_2 - \frac{3}{2}).$$

8.4.3 Analyzing the Non-Outer-Faces (Proof of Lemma 8.36)

We split the proof of Lemma 8.36 into three parts. First, we show an upper bound for the number of surviving triangles if a super-face f has $|E(f)| > 3$ or $a_1^{free}(f) + p_0^{base}(f) > 0$. Then we show that $survive(f) \leq \mu(f) - \frac{3}{2}$, if $|E(f)| = 3$ and $a_1^{free}(f) + p_0^{base}(f) = 0$. Finally, we combine both results to give the upper bound for the number of surviving triangles in each type of super-face in $\mathcal{F}$.

Lemma 8.38. *Let $f \in \mathcal{F}$, if $|E(f)| > 3$ or $a_1^{free}(f) + p_0^{base}(f) > 0$ we have*

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - 2.$$

Proof. If $|E(f)| = 3$ and $a_1^{free}(f) + p_0^{base}(f) \geq 1$, it is easy to enumerate all possible compositions of the face boundary of f and check for each case that the claimed bound holds.

- ($a_1^{free}(f)+p_0^{base}(f) = 1$:) In this case, $survive(f) = 0$, $|Free(f)| = 1$ and $|Occ(f)| = 2$.

- ($a_1^{free}(f) + p_0^{base}(f) = 2$:) In this case, $survive(f) = 0$, and $|Free(f)| = 2$.

- ($a_1^{free}(f) + p_0^{base}(f) = 3$:) In this case, $survive(f) = 1$, $|Free(f)| = 3$, and $|Occ(f)| = 0$.

Now consider the case where $|E(f)| > 3$. To bound $survive(f)$ in this case, we *locally* modify the internal structure for a fixed f in a special way. Notice that we make these modifications only for counting purposes and they do not change the structure of our graph G in any way. First, we *decouple* the supported cross triangles embedded inside f which share their landing components by adding a dummy landing vertex for each such cross triangle and making the new dummy vertex its landing component. Then using additional type-0 edges we triangulate the super-face f in an arbitrary way. Note that the decoupling step allows us to get a full triangulation for f and at the same time this operation does not reduce the value of $survive(f)$ for f (see Figure 8.15 for illustration). Hence, any bound which we get after performing this operation also holds for the original quantity $survive(f)$. This triangulation of the super-face f has exactly $|E(f)| - 2$ triangular faces. Starting with this bound, we use the particular structure of f to achieve the desired bound for $survive(f)$.

By Observation 8.35 no edge of type-2, occupied side of a type-1 edge or base side of a type-1 triangle can be adjacent to any triangular face in $survive(f)$. Also, at most two of these edges could belong to any triangular face in f. Hence, out of all the potential $|E(f)| - 2$ faces in the triangulate super-face f, at least $\left\lceil \frac{|Occ(f)|}{2} \right\rceil$ faces will be killed and hence we get the claimed bound on $survive(f)$. □

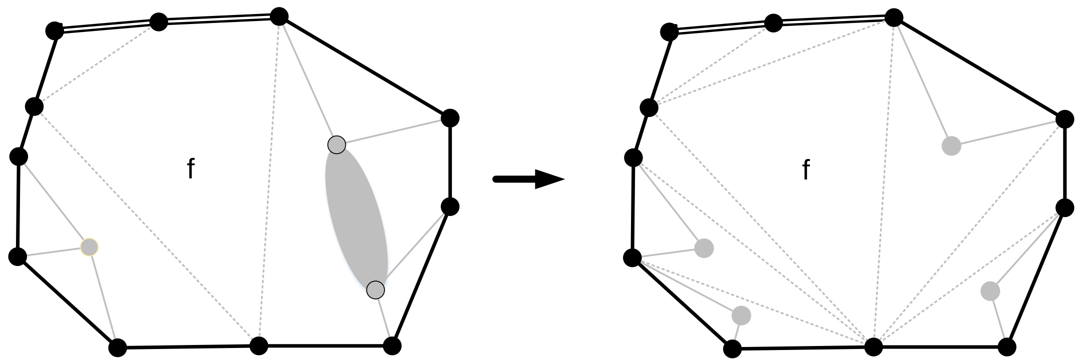


Figure 8.15: The decoupling and triangulation operation for a super-face $f \in \mathcal{F}$. Notice that we make these modifications only for counting purposes and that they maintain the structure of our original graph G.

For some other cases, we can still get a slightly weaker bound.

Lemma 8.39. *Otherwise, if* $|E(f)| = 3$ *and* $a_1^{free}(f) + p_0^{base}(f) = 0$, *then we have*

$$survive(f) \leq \mu(f) - \frac{3}{2}.$$

Proof. Notice that $|E(f)| = 3$ implies $p^{free}(f) = 0$. Hence, the first inequality is trivially true by substituting the value $a_1^{occ}(f) + a_2(f) + p_1^{base}(f) = 3$ and $2p^{free}(f) + a_1^{free}(f) + p_0^{base}(f) = 0$. □

Now, we are ready to complete the proof of Lemma 8.36. For any type-$[1, 0]$ super-face f, $|Occ(f)| = 1$ and $|E(f)| > 3$, hence using Lemma 8.38, we get

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - 2 = |Free(f)| + \frac{|Occ(f)|}{2} - \frac{5}{2} = \mu(f) - \frac{5}{2}.$$

For any type-$[1, 1]$ or type-$[2, \bullet]$ super-face f we have that $|E(f)| > 3$, hence by Lemma 8.38, we get

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - 2 \leq |Free(f)| + \frac{|Occ(f)|}{2} - 2 = \mu(f) - 2.$$

For any type-$[\geq 3, \bullet]$ super-face f, if $|Occ(f)| = |E(f)| = 3$, then Lemma 8.39 implies

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - \frac{3}{2} \leq |Free(f)| + \frac{|Occ(f)|}{2} - \frac{3}{2} = \mu(f) - \frac{3}{2}.$$

Otherwise using Lemma 8.38 we get

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - 2 \leq |Free(f)| + \frac{|Occ(f)|}{2} - 2 = \mu(f) - 2.$$

8.5 A Classification Scheme for Factor Six

The classification scheme of the super-faces in H given in Section 8.4 did not take advantage of the fact yet that $\mathcal{C}[S]$ is a 2-swap optimal cactus subgraph of G. We will show a classification scheme that certifies the factor six bound by extending the classification scheme of Section 8.4 by super-face types that heavily exploit this fact. The important observation that leads to a better bound is to derive a better gain for super-faces of type-$[1,\bullet]$ and type-$[2,\bullet]$ from the previous classification. We notice that, for a certain sub-class of these super-faces, a better bound can be obtained.

A New Super-face Classification: Now we sub-categorize type-$[1,\bullet]$ and type-$[2,\bullet]$ super-faces into further classes, based on the values of $a_1^{free}(f)$ and $p_0^{base}(f)$. A super-face f will be of type-$[i,j,k]$ if $p_1^{base}(f)+a_2(f)+a_1(f)=i$, $a_1^{free}(f)=j$ and $p_0^{base}(f)=k$. If there is no restriction on a particular dimension, then we put a dot ($[\bullet]$) there. Following is the categorization of super-faces which we use.

- type-$[1,\bullet,\bullet]$: $p_1^{base}(f)+a_2(f)+a_1(f)=1$.
  - type-$[1,0,\bullet]$: $a_1^{free}(f)=0$.
    * type-$[1,0,0]$: $p_0^{base}(f)=0$.
    * type-$[1,0,\geq 1]$: $p_0^{base}(f)\geq 1$.
  - type-$[1,1,\bullet]$: $a_1^{free}(f)=1$.
    * type-$[1,1,0]$: $p_0^{base}(f)=0$.
    * type-$[1,1,\geq 1]$: $p_0^{base}(f)\geq 1$.
- type-$[2,\bullet,\bullet]$: $p_1^{base}(f)+a_2(f)+a_1(f)=2$.
  - type-$[2,0,\bullet]$: $a_1^{free}(f)=0$.
    * type-$[2,0,0]$: $p_0^{base}(f)=0$.
    * type-$[2,0,\geq 1]$: $p_0^{base}(f)\geq 1$.
  - type-$[2,1,\bullet]$: $a_1^{free}(f)=1$.
  - type-$[2,2,\bullet]$: $a_1^{free}(f)=2$.
- type-$[\geq 3,\bullet,\bullet]$: $p_1^{base}(f)+a_2(f)+a_1(f)\geq 3$.

Let the subset $\mathcal{F}[i,j,k]\subseteq\mathcal{F}$ be the set of type-$[i,j,k]$ super-faces and analogously let $\eta[i,j,k]=|\mathcal{F}[i,j,k]|$. It is easy to see that the categorization partitions the set $\mathcal{F}\setminus\{f_0\}$, $\mathcal{F}[i,j,k]\subseteq\mathcal{F}[i,j,\bullet]\subseteq\mathcal{F}[i,\bullet,\bullet]$ for any i,j,k, which implies, $|\mathcal{F}|=1+\eta[1,\bullet,\bullet]+\eta[2,\bullet,\bullet]+\eta[\geq 3,\bullet,\bullet]$. Also, $\eta[i,\bullet,\bullet]=\sum_j\eta[i,j,\bullet]$ for each i, $\eta[i,j,\bullet]=\sum_k\eta[i,j,k]$ for each i,j.

We classify a sub-class of type-$[1,0,0]$, type-$[1,1,0]$, and type-$[2,0,0]$ super-faces that admits an improved bound via several new notions.

Adjacent triangles, edges and friends: Let t_1 and t_2 be two cactus triangles that share a vertex. Denote their vertices by $V(t_i) = \{u_i, v_i, w_i\}$, where $v_1 = v_2$ (say v). In this case, we call them *adjacent triangles.* Let w_i be a free vertex of t_i. If there is a way to embed an edge w_1w_2 such that the region bounded by (v, w_1, w_2) is empty, we say that these triangles are *strongly adjacent* (see Figure 8.16 for an example).

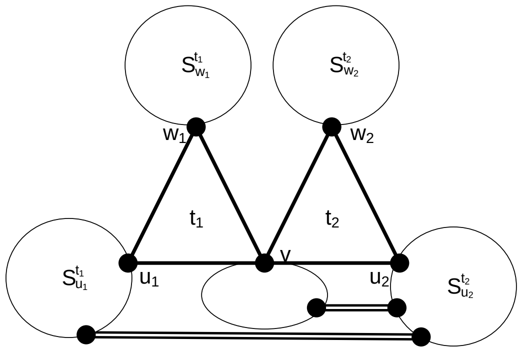


Figure 8.16: An example of two *strongly*-adjacent triangles in H.

Otherwise, the two triangles are called *weakly adjacent* (as shown in Figure 8.17. Furthermore, if t_1 and t_2 are strongly adjacent in H and $w_1w_2 \in E(G[S])$, then we say that t_1 and t_2 are *friends* or friendly triangles (as depicted in Figure 8.18.

Observation 8.40. *The free sides for any pair of triangles that are strongly-adjacent or friends are part of the same super-face in $\mathcal{F}$.*

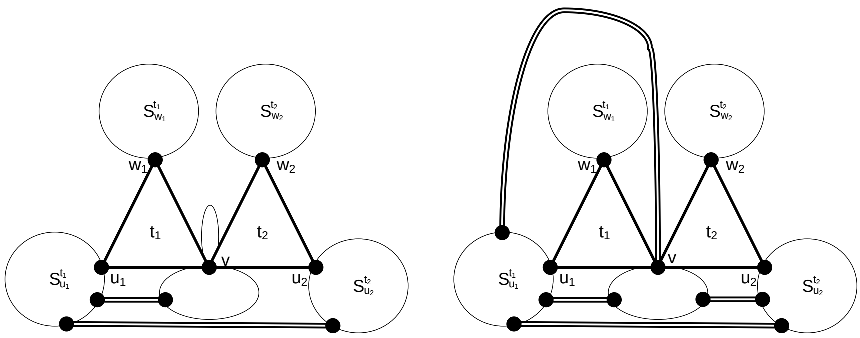


Figure 8.17: Two examples where two triangles in H are *weakly*-adjacent.

We will crucially rely on the following lemma, whose proof is provided later in Subsection 8.5.4

Lemma 8.41 (Friend Lemma)**.** *The following properties hold:*

- *No type-*1 *heavy triangle is friends with any other heavy cactus triangle.*
- *For any pair of type-*0 *triangles which are friends, their corresponding base sides belong to a common super-face in $\mathcal{F}$.*

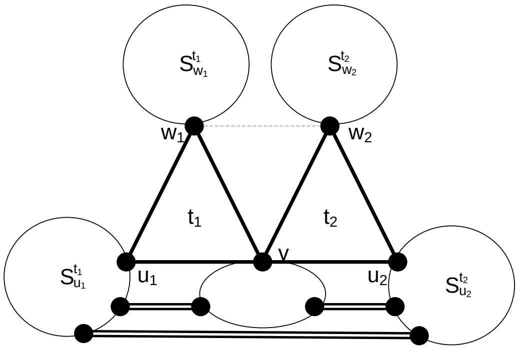


Figure 8.18: Two adjacent triangles which are *friends* in $G[S]$. In H these two triangles will be *strongly*-adjacent.

As Lemma 8.41 states that no type-1 triangle in $\mathcal{C}[S]$ is friends with another triangle, from hereon, whenever we argue about friends, we always refer to a pair of type-0 triangles.

Friendly super-faces: We call a super-face $f \in \mathcal{F}$ of type-$[1,0,0]$, $[1,1,0]$ or $[2,0,0]$ a *friendly* super-face if it contains at least one pair of cactus triangles that are friends. Let $\mathcal{F}_{fri}[1,0,0] \subseteq \mathcal{F}[1,0,0]$, $\mathcal{F}_{fri}[1,1,0] \subseteq \mathcal{F}[1,1,0]$ and $\mathcal{F}_{fri}[2,0,0] \subseteq \mathcal{F}[2,0,0]$ be the set of friendly super-faces of type-$[1,0,0]$, $[1,1,0]$ and $[2,0,0]$ respectively. Also, let $\eta_{fri}[i,j,k] = |\mathcal{F}_{fri}[i,j,k]|$. Let $\eta_{fri} = \eta_{fri}[1,0,0] + \eta_{fri}[1,1,0] + \eta_{fri}[2,0,0]$.

The following lemmas (which are proven in later subsections) give us stronger bounds on $survive(f)$ for super-faces of type-$[1,0,0]$, $[1,1,0]$ or $[2,0,0]$ which are not friendly.

Lemma 8.42. *For any type-*$[1,0,0]$ *super-face* $f \in \mathcal{F}[1,0,0] \setminus \mathcal{F}_{fri}[1,0,0]$*, the following bound holds for gain*(f)*.*

$$gain(f) \geq \frac{9}{2}.$$

Lemma 8.43. *For any type-*$[1,1,0]$ *super-face* $f \in \mathcal{F}[1,1,0] \setminus \mathcal{F}_{fri}[1,1,0]$*, the following bound holds for survive*(f)*.*

$$gain(f) \geq 4.$$

Lemma 8.44. *For any type-*$[2,0,0]$ *super-face* $f \in \mathcal{F}[2,0,0] \setminus \mathcal{F}_{fri}[2,0,0]$*, the following bound holds for survive*(f)*.*

$$gain(f) \geq 3.$$

We have successfully identified a set of super-faces for which we obtain an improved bound. For the remaining super-faces, we will rely on trivial upper bounds.

Lemma 8.45. *For any super-face* $f \in \mathcal{F}$*, the respective bounds hold for gain*(f)

- *type-*$[1,0,\geq 1]$*:*

$$gain(f) \geq \frac{5}{2}.$$

- *(*$f \in \mathcal{F}_{fri}[1,0,0]$*):*

$$gain(f) \geq \frac{5}{2}.$$

- *type*-$[1,1,\geq 1]$*:*
$$gain(f) \geq 2.$$
- *($f \in \mathcal{F}_{fri}[1,1,0]$):*
$$gain(f) \geq 2.$$
- *type*-$[2,0,\geq 1]$*:*
$$gain(f) \geq 2.$$
- *($f \in \mathcal{F}_{fri}[2,0,0]$):*
$$gain(f) \geq 2.$$
- *type*-$[2,1,\bullet]$*:*
$$gain(f) \geq \frac{5}{2}.$$
- *type*-$[2,2,\bullet]$*:*
$$gain(f) \geq 2.$$
- *type*-$[\geq 3,\bullet,\bullet]$*:*
$$gain(f) \geq \frac{3}{2}.$$

Proof. For any type-$[1,0,\bullet]$ or type-$[2,1,\bullet]$ super-face f, $|Occ(f)| = 1$ and $|E(f)| > 3$, hence using Lemma 8.38, we get

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - 2 = |Free(f)| + \frac{|Occ(f)|}{2} - \frac{5}{2} = \mu(f) - \frac{5}{2}.$$

For any type-$[1,1,\bullet]$ or type-$[2,0,\bullet]$ or type-$[2,2,\bullet]$ super-face f, $|E(f)| > 3$, hence using Lemma 8.38, we get

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - 2 \leq |Free(f)| + \frac{|Occ(f)|}{2} - 2 = \mu(f) - 2.$$

For any type-$[\geq 3,\bullet,\bullet]$ super-face f, if $|Occ(f)| = |E(f)| = 3$, using Lemma 8.39, we get

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - \frac{3}{2} \leq |Free(f)| + \frac{|Occ(f)|}{2} - \frac{3}{2} = \mu(f) - \frac{3}{2}.$$

Else, using Lemma 8.38, we get

$$survive(f) \leq |Free(f)| + \left\lfloor \frac{|Occ(f)|}{2} \right\rfloor - 2 \leq |Free(f)| + \frac{|Occ(f)|}{2} - 2 = \mu(f) - 2.$$

□

8.5.1 Valid Inequalities

We present various upper bounds on the number of super-faces of a certain type. We denote by Φ the following system of linear inequalities.

Lemma 8.46 (Various upper bounds on the number of super-faces)**.** *The following bounds hold:*

- $\eta[2,\bullet,\bullet] + 2\eta[\geq 3,\bullet,\bullet] \leq p_1 + |\mathcal{F}| - 2$.
- $\eta[1,1,\bullet] + \eta[2,1,\bullet] + 2\eta[2,2,\bullet] \leq a_1$.
- $\eta_{fri} + \eta[1,0,\geq 1] + \eta[1,1,\geq 1] + \eta[2,0,\geq 1] \leq p_0$.

Proof. The first bound is derived in the same manner as in Lemma 8.37. The second bound is also similar. Consider the sum:

$$\sum_{f \in \mathcal{F}[1,1,\bullet] \cup \mathcal{F}[2,1,\bullet] \cup \mathcal{F}[2,2,\bullet]} a_1^{free}(f) \leq a_1.$$

Notice that each super-face of type-$[1,1,\bullet]$ or type-$[2,1,\bullet]$ gets the contribution of at least one, while the other type gets the contribution of two, thus we have that the sum is at least $\eta[1,1,\bullet] + \eta[2,1,\bullet] + 2\eta[2,2,\bullet]$.

Finally, for the third bound, we give a combinatorial charging argument. First, we imagine giving one unit of money to each type-0 triangle. Therefore, p_0 units of money are placed into the system. We will argue that we can "transfer" this amount such that each super-face in $\mathcal{F}_{fri}[1,0,0] \cup \mathcal{F}_{fri}[1,1,0] \cup \mathcal{F}_{fri}[2,0,0] \cup \mathcal{F}[1,0,\geq 1] \cup \mathcal{F}[1,1,\geq 1] \cup \mathcal{F}[2,0,\geq 1]$ receives at least one unit of money, hence establishing the desired bound.

- For each face $f \in \mathcal{F}_{fri}[1,0,0] \cup \mathcal{F}_{fri}[1,1,0] \cup \mathcal{F}_{fri}[2,0,0]$, we know that there must be at least one pair of friends. By Lemma 8.41, no type-1 triangle is friends with any other heavy cactus triangle. The super-face f receives one unit of money from each such triangle in the pair, thus we have two units on each such super-face.
- Now consider a super-face $f \in \mathcal{F}[1,0,\geq 1] \cup \mathcal{F}[1,1,\geq 1] \cup \mathcal{F}[2,0,\geq 1]$. On such super-face, there is at least one type-0 triangle, and such cactus triangle would (i) pay super-face f if it still has the money, or (ii) the "extra" money would be put in the system to pay f if no cactus triangle in f has money left with it.

In the end, all such super-faces would have at least one or two units of money, thus the total money in the system is at least $2\eta_{fri} + \eta[1,0,\geq 1] + \eta[1,1,\geq 1] + \eta[2,0,\geq 1]$. The total payment into the system is at most p_0 plus the extra money. There can be at most η_{fri} units of extra money spent: Due to Lemma 8.41, i.e. whenever a face contains a triangle that spent in the first step, it must also contain its pair of friends, thus there can be at most η_{fri} such faces that cause an extra spending. This reasoning implies that

$$2\eta_{fri} + \eta[1,0,\geq 1] + \eta[1,1,\geq 1] + \eta[2,0,\geq 1] \leq p_0 + \eta_{fri}.$$

□

Deriving the factor six: Now that we have both the inequalities and the gain bounds, the following is an easy consequence (e.g. it can be verified by an LP solver). For completeness, we produce a human-verifiable proof in Subsection 8.5.5.

Lemma 8.47.

$$q \leq 4p + \frac{1}{2}p_1 + \frac{5}{2}a_1 + 3a_2 - \overrightarrow{gain} \cdot \vec{\chi} \leq 6p - \phi(S).$$

8.5.2 Gain Analysis for Other Cases

In this subsection, we analyze the gain for various types of super-faces of H where we get improved bounds over the types used in Section 8.4.

Analyzing Super-Faces of Type $\mathcal{F}[1,0,0] \setminus \mathcal{F}_{fri}[1,0,0]$ (Proof of Lemma 8.42)

A super-face in this set turns out to behave in a very structured way, i.e., the edges of the cactus triangles bounding this face look like a "fence", which is made precise below.

Cactus fence: A *cactus fence* of size k is a maximal sequence of cactus triangles $(t_1, \ldots, t_k)$ such that any pair t_i and t_{i+1} are strongly adjacent. Moreover, for each triangle t, if $w \in V(t)$ is a free vertex of t, then S_w^t is a singleton.

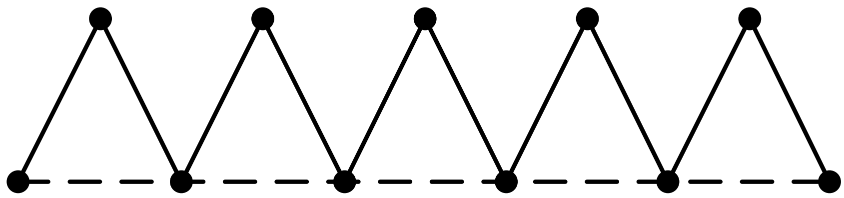


Figure 8.19: A cactus fence structure of size five.

Lemma 8.48 (Fence lemma)**.** *Any super-face $f \in \mathcal{F}[1,0,0] \setminus \mathcal{F}_{fri}[1,0,0]$ is bounded by free sides of a cactus fence together with one edge e that is of type-*2*.*

The proof of this lemma is quite intricate and is deferred to the upcoming Subsection 8.5.3. Moreover, from the definition of the set $\mathcal{F}[1,0,0] \setminus \mathcal{F}_{fri}[1,0,0]$, each pair of cactus triangles on this face is not a pair of friends. It suffices to show that $survive(f) \leq |E(f)| - 5$: Since $|Occ(f)| = 1$, this will imply $survive(f) \leq |E(f)| - 5 = |Free(f)| + |Occ(f)| - 5 = |Free(f)| + |Occ(f)|/2 - \frac{9}{2} = \mu(f) - \frac{9}{2}$ which proves Lemma 8.42.

For obtaining the bound on $survive(f)$, we construct an auxiliary graph H' on $V(f)$ by modifying the inside of the super-face f. First, we decouple the supported cross triangles embedded inside f which share their landing components by adding a dummy landing vertex for each such cross triangle and making the new dummy vertex its landing component. Then the inside of f is fully triangulated using additional type-0 edges such that in total it contains $|E(f)| - 2$ triangular faces. Notice that, this process cannot decrease the number of $survive(f)$ triangles embedded inside of f in H'.

Lemma 8.49. *If a super-face $f : |E(f)| \geq 5$ contains a single cactus fence structure and only one additional edge, then any triangulation of f using type-0 edges must contain the free sides for at least one pair of cactus triangles which are friends.*

Proof. The lemma follows easily using the facts that in any triangulation of a polygon there are at least two triangles each containing two sides of the polygon and no two base vertices can be joined by an edge inside super-face f as this will create a multi-edge, hence there should be at least one triangular face containing two adjacent free edges each belonging a different cactus triangle from a pair of strongly adjacent cactus triangles. □

It is clear that $|E(f)| \geq 5$, hence by Lemma 8.49, H' contains an edge e' joining strongly adjacent pair of cactus triangles. Hence, $e' \in E(H')$ but not embedded inside f in G (since G cannot contain any pair of friends), thus $H' \setminus e'$ still contains all surviving faces in the original graph and has only $|E(f)| - 4$ triangular faces inside f. Since the friends edge e' goes across the two free vertices of two cactus triangles, but e joins two base vertices of two cactus triangles, hence they cannot form a triangle together. This implies at least one more triangular face which is bounded by e, does not survive, which proves Lemma 8.42.

Analyzing Super-Faces of Type $\mathcal{F}[1,1,0] \setminus \mathcal{F}_{fri}[1,1,0]$ (Proof of Lemma 8.43)

We can use the same reasoning as in the proof of Lemma 8.42 to prove Lemma 8.43. The only difference is that $a_2(f)+a_1(f)+p_1^{base}(f) = 1$ and $a_1^{free}(f) = 1$ implies $|Occ(f)| = 0$. We can show $survive(f) \leq |E(f)| - 4 = \mu(f) - 4$ by simply using the absence of the edge e' (from Lemma 8.49), and therefore the missing of two surviving faces from the triangulation in the interior of the super-face f.

Analyzing Super-Faces of Type $\mathcal{F}[2,0,0] \setminus \mathcal{F}_{fri}[2,0,0]$ (Proof of Lemma 8.44)

Here it suffice to show that $survive(f) \leq |E(f)| - 4$: As $|Occ(f)| = 2$, this implies that $survive(f) \leq |E(f)|-4 = |Free(f)|+|Occ(f)|-4 = |Free(f)|+|Occ(f)|/2-3 = \mu(f)-3$ which proves the lemma.

Similarly, to the proof of Lemma 8.42, let H' be the maximal auxiliary graph on $V(f)$ that contains all edges embedded in the interior of f in G. Then H' has $|E(f)| - 2$ triangular faces inside of f. Since $p_1^{base}(f) + a_2(f) + a_1^{occ}(f) = 2$, let e_1 and e_2 be the two edges bounding f that contribute to this sum. If e_1 and e_2 bound different faces of H', then we are done since the number of surviving faces of H' is at most $|E(f)| - 4$.

Now, assume that e_1 and e_2 bound the same face of H'. We give the proof of the following lemma in Subsection 8.5.3.

Lemma 8.50 (The second fence lemma)**.** *For any super-face $f \in \mathcal{F}[2,0,0] \setminus \mathcal{F}_{fri}[2,0,0]$, if the two edges corresponding to $p_1^{base}(f) + a_2(f) + a_1(f)$ are adjacent, then the face consists of a cactus fence of size $p^{free}(f)$ together with two edges e_1 and e_2 that contribute to the sum $p_1^{base}(f) + a_2(f) + a_1(f)$.*

Since, both e_1 and e_2 bound the same triangular face of H', they must be adjacent. Let e be the third edge which bounds the triangular face adjacent to both e_1 and e_2 embedded inside of f in H'. Now, consider the graph $\tilde{H} = H' \setminus \{e_1, e_2\}$, thus $\tilde{H}$ consists of a cactus fence together with e. Using Lemma 8.49, $\tilde{H}$ must contain an edge joining a strongly adjacent pair of cactus triangles. This edge cannot exist in the original graph since f contains no pair of friends, thus $\tilde{H} \setminus e$ still contains all surviving faces of the original graph. But it contains at most $|E(f)| - 5$ surviving faces.

8.5.3 Proving the Fence Lemmas

In this section, we prove the two fence lemmas used in deriving the gain bounds in the previous section. An important notion that we will use is that of the *trapped triangles.*

Trapped and free triangles: We further classify heavy cactus triangles based on whether their free component is a singleton or not. Let f be a face that contains free sides of heavy triangle t. If the free component of heavy triangle t is a singleton, then we call t a *free* triangle, else it will be a *trapped* triangle inside f.

The following lemma is a generalization of both Lemma 8.48 and Lemma 8.50, which we used in the previous subsections.

Lemma 8.51. *For any super-face $f \in \mathcal{F}$ with $p^{base}(f) = 0$, if $a_1(f) + a_2(f) = 1$ or if $a_1(f) + a_2(f) = 2$ but the two type-1 and type-2 edges are adjacent:*

- *Then, there can be no triangle trapped inside f and*
- *Every pair of adjacent triangles is strongly adjacent.*

Proof. We argue in two steps. First we show that every triangle is not trapped inside f. Assume otherwise, that some $t : V(t) = \{u, v, w\}$ is trapped, and the free component S^t_w is not a singleton. Since S^t_w is a free component, we have that $B^t_{wu} \cup B^t_{wv}$ is empty. By using Observation 8.34 on S^t_w, there is at least one type-1 or type-2 edge, say e, bounding the outer-face of the graph $H[S^t_w]$ and edge e also bounds the face f (see Figure 8.20).

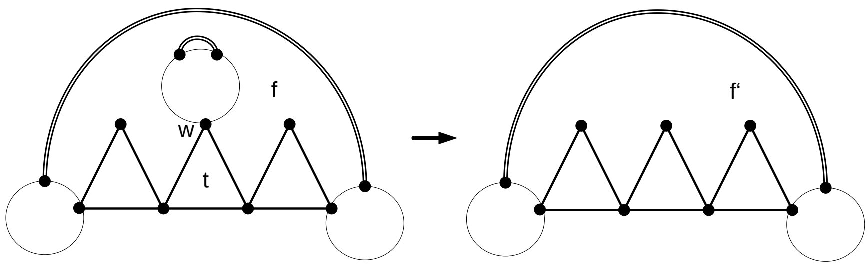


Figure 8.20: The contraction operation when f contains a trapped triangle's free side.

Now consider the contracted graph that contracts S^t_w into a single vertex. Let f' be the residual super-face corresponding to f and S' be the residual component after the contraction of S^t_w. Notice that, the graph $H[S']$ contains only heavy triangles: For any cactus triangle t' in $H[S']$, no type-1 or type-2 edge that contributes to its "heaviness" was contracted. This implies that the super-face f' of $H'[S']$ contains at least one type-1 or type-2 edge, say e' (by Observation 8.34). It is easy to verify that e and e' are not adjacent.

Next we prove the second property. Let t_1 and t_2 be an adjacent pair of triangles whose free sides bound the super-face f. We will argue that $t_1 : V(t_1) = \{u_1, v_1, w_1\}$ and $t_2 : V(t_2) = \{u_2, v_2, w_2\}$ are strongly adjacent, with w_i being the free vertex of t_i and $v_1 = v_2$ being the common vertex. Assume that they were not strongly adjacent. Notice that, since the free sides for both t_1 and t_2 bound a common super-face f, this can only happen if $S^{t_1}_{v_1}$ has a connected component $S' : S' \subseteq S^{t_1}_{v_1} \cap S^{t_2}_{v_2}$ embedded inside f (see Figure 8.21). Observe that $C[S']$ contains only heavy cactus triangles: Any type-1 or type-2 edge with exactly one endvertex in S' can only be incident to v_1 and must be embedded in the exterior of f. Again, as in the previous case, we can do the contraction trick to argue that there exist two type-1 or type-2 edges e and e' bounding face f such that $e \neq e'$ and they are not adjacent. □

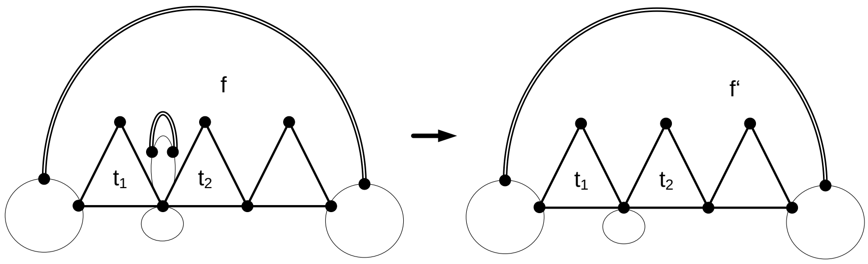


Figure 8.21: The contraction operation when f contains a pair of free sides which corresponds to a pair of weakly-adjacent triangles.

8.5.4 Proof of the Friend Lemma (Proof of Lemma 8.41)

We now present the proof of Lemma 8.41. We will rely on the three following structural observations.

Lemma 8.52. *Let $f \in \mathcal{F}$ and $t = (u, v, w)$ be any heavy cactus triangle such that $E(t) \cap E(f) \neq \emptyset$, and $uv \in E(t)$ be its (unique) cactus edge for which $B^t_{uv} \neq \emptyset$. Then, we have $|E(f) \cap B^t_{uv}| = 1$.*

Proof. Let P_2 be a maximal trail along the boundary of f starting from u and only visiting vertices in S^t_u in graph H. Notice that P_2 may use cactus edges or type-1 or type-2 edges. Let u_2 be the other endvertex of P_2 and u_2u_3 be the next edge on the boundary of f, such that $u_3 \in S^t_v \cup S^t_w$. First, notice that u_3 cannot be in S^w_t , for otherwise, we would have the free sides of t on different super-faces. Therefore, $u_3 \in S^t_v$. Now, let P_3 be a maximal trail from u_3 along the boundary of f, visiting only vertices in S^t_v. We claim that P_3 must contain v: Otherwise, let v' be the last vertex on P_3 and e' be the next edge on f incident to v'. Consider a region R bounded by (i) the sides of t on super-face f,(ii) trail $P_2u_3P_3$,and (iii) any path from v' to v using only cactus edges in S^t_v. This close region must contain super-face f, thus e' must be embedded inside R (see Figure 8.22). This is a contradiction since e' cannot connect v' to a vertex in S^t_w (same reasoning as before), and similarly it cannot connect v' to S^t_u (this would contradict the choice of u_2 or the edge u_2u_3). □

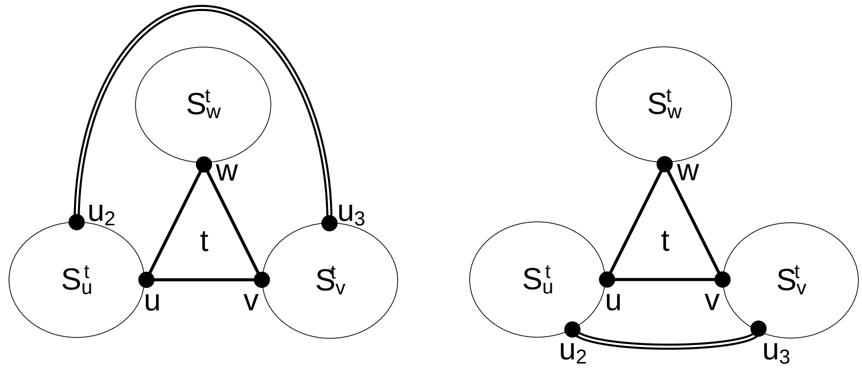


Figure 8.22: An illustration of the regions containing the free sides or base sides of a heavy triangle t in the proof of Lemma 8.52.

Observation 8.53. *For any heavy triangle t, the free and the base edges will be adjacent to two different super-faces in $\mathcal{F}$.*

Let t be a heavy triangle. Let $f, f' \in \mathcal{F}$ be the two different super-faces from Observation 8.53, that contain the base and free edges of t respectively. Then we can show the following.

Lemma 8.54. *Let e, e' be the unique type-1 or type-2 edges on f and f' across the occupied components of t (which must exist by Lemma 8.52). Then $e \neq e'$.*

Proof. Assume otherwise that $e = e'$, thus the super-faces f and f' are adjacent at e. This means that there is only one type-1 or type-2 edge across the occupied components, contradicting the fact that t is heavy (see Figure 8.23 for an illustration). ☐

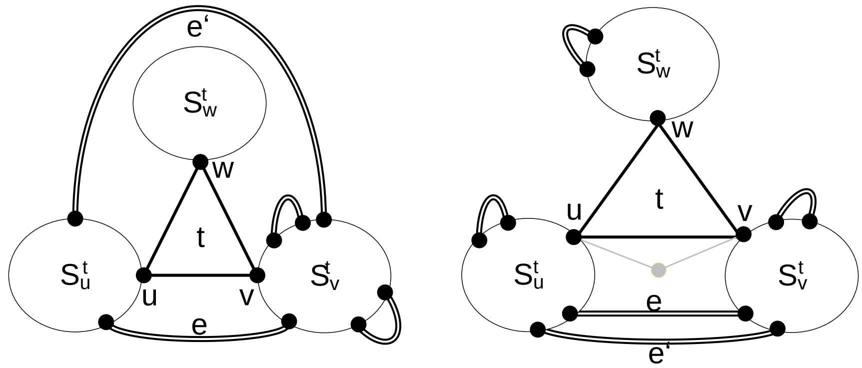


Figure 8.23: Two possible compositions of super-faces containing the free and base edges of a heavy triangle t.

Components for two adjacent heavy triangles: Now we fix the labeling for the new components created by the operation of removing edges for two adjacent heavy triangles from $\mathcal{C}[S]$, which we will use in the rest of this section. Every time when we argue about two adjacent heavy triangles we will denote them by t_1 and t_2 such that $V(t_1) = \{u_1, v_1 = v, w_1\}$ and $V(t_2) = \{u_2, v_2 = v, w_2\}$, where w_1, w_2 will be the corresponding free vertices and v the common base vertex of t_1 and t_2. The vertices of the new components formed by removing edges $E(t_1) \cup E(t_2)$ from $\mathcal{C}[S]$ will be $S^{t_1}_{w_1}, S^{t_2}_{w_2}, S^{t_1}_{u_1}, S^{t_2}_{u_2}, S_v$, such that $w_1 \in S^{t_1}_{w_1}, w_2 \in S^{t_2}_{w_2}, u_1 \in S^{t_1}_{u_1}, u_2 \in S^{t_2}_{u_2}$ and $v \in S_v$. Notice that the free components of t_1 and t_2 are $S^{t_1}_{w_1}, S^{t_2}_{w_2}$ respectively, the occupied components of t_1 are $S^{t_1}_{u_1}, S^{t_1}_{v_1} = S_v \cup S^{t_2}_{w_2} \cup S^{t_2}_{u_2}$ and the occupied components of t_2 are $S^{t_2}_{u_2}, S^{t_2}_{v_2} = S_v \cup S^{t_1}_{w_1} \cup S^{t_1}_{u_1}$.

Lemma 8.55. *Let $f \in \mathcal{F}$ be a super-face. Let $t_1, t_2 : V(t_i) = (u_i, v_i, w_i)$ be two adjacent heavy cactus triangles with $V(t_1) \cap V(t_2) = v_1 = v_2$ (say v) such that $E(t_i) \cap E(f) \neq \emptyset$ for $i \in \{1, 2\}$. For each i, let $u_iv_i \in E(t_i)$ be the base edge and $B^{t_i}_{u_iv_i} \cap E(f) = \{e_i\}$ (unique due to lemma 8.52).*

(1) $e_1 = e_2 := e$ if and only if the common edge e goes across $S^{t_1}_{u_1}$ and $S^{t_2}_{u_2}$.

(2) $e_1 \neq e_2$ if and only if both e_1, e_2 are incident to S_v.

Proof. The first direction for item (1) is easy to see by the way $S^{t_1}_{u_1}, S^{t_2}_{u_2}$ are defined and by the fact that e goes across the occupied components for both t_1 and t_2. In the other direction, if e_1 goes across $S^{t_1}_{u_1}, S^{t_2}_{u_2}$, hence it also goes across the occupied components $S^{t_2}_{u_2}, S^{t_2}_{v_2}$ for t_2. This along with the fact that e_1 belongs to f and Lemma 8.52, it implies that $e_2 = e_1$.

One direction for item (2) follows from the negation of item (1) because if any one of e_1 or e_2 goes across $S^{t_1}_{u_1}, S^{t_2}_{u_2}$, then it implies $e_1 = e_2$. On the other hand, if one of e_1 or e_2 is incident to S_v, then they cannot be same by item (1). □

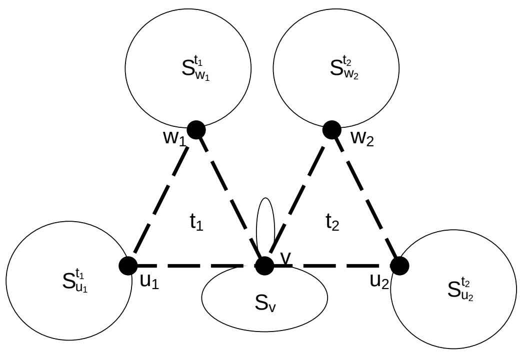


Figure 8.24: The structure of split-components formed by removing two adjacent triangles.

Let $f \in \mathcal{F}$ be a super-face. Let $t_1, t_2 : V(t_i) = (u_i, v_i, w_i)$ be two adjacent heavy cactus triangles with $v_1 = v_2$ (say v) such that both of whose free sides belong to f. Then we can show the following.

Lemma 8.56. *If the base sides for t_1 and t_2 belong to two different super-faces $f_1, f_2 \in \mathcal{F}$ and for each i, let $B^{t_i}_{u_iv_i} \cap E(f_i) = \{e_i\}$ (unique due to lemma 8.54). Then at least one of e_1 or e_2 is incident to some vertex in S_v, which in turn implies $e_1 \neq e_2$.*

Proof. As t_1 and t_2 are adjacent, we use the notations defined above for the various components corresponding to two adjacent heavy triangles. First, assuming that at least one of e_1 or e_2 is incident to some vertex in S_v, we prove that $e_1 \neq e_2$.

By contradiction, let $e_1 = e_2 := u'v'$. Now we show that there will be a cycle in $\mathcal{C}[S]$ sharing an edge with t_1, contradicting the fact that $\mathcal{C}$ is a triangular cactus (see Figure 8.25). By the above claim, this edge is incident to S_v (say $v' \in S_v$). Also, by the way e_1 and e_2 are defined, the other endvertex u' belongs to both $S^{t_1}_{u_1}$ and $S^{t_2}_{u_2}$. Hence, in $\mathcal{C}[S] \setminus (E[t_1] \cup E[t_2])$, using only cactus edge, there is a path P_1 from u' to u_1 and another path P_2 from u' to u_2. Hence, $u'P_1u_1 \cup u_1v \cup vu_2 \cup u_2P_2u'$ is a cycle in $\mathcal{C}[S]$ sharing edge u_1v with t_1, contradicting the fact that $\mathcal{C}$ is a triangular cactus.

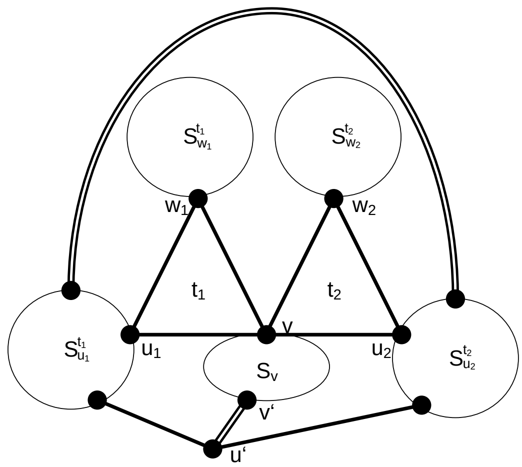


Figure 8.25: An illustration of the argument used to reach a contradiction in the proof of Lemma 8.56. There exists a cycle in $\mathcal{C}[S]$ sharing the edge u_1v with t_1, if we assume $e_1 = e_2 := u'v'$.

Finally, we prove that at least one of e_1 or e_2 is incident to some vertex in S_v. For contradiction assume none of e_1, e_2 are incident to S_v. Notice that since free sides of t_1 and t_2 belong to the same super-face f, we can partition the component S_v into two parts S'_v, S''_v (with an exception that v is a common vertex), such that S'_v is embedded inside f, S''_v outside of f and v lies on f. Starting with vertex u_1 and the base side u_1v create a maximal trail P_1 in H along the boundary of f_1 by visiting type-1, type-2 or cactus edges and vertices only from S''_v. This trail should end at some vertex $v' \in S_v$ (possibly v), such that there is a type-1 or type-2 edge leaving S''_v incident to v' (say $v'u'$). If not, then the trail would end at v and the base side vu_2 will be the next edge belonging to super-face f_1 in the graph H, which contradicts our assumption. Notice that since $S^{t_1}_{w_1}, S^{t_2}_{w_2}$ are the free components, hence either $u' \in S^{t_1}_{u_1}$ or $u' \in S^{t_2}_{u_2}$. In case when $u' \in S^{t_1}_{u_1}$, it implies that $v'u'$ is an edge going across the components of t_1 and also belongs to f_1. But by Lemma 8.52, it implies that $e_1 = u'v'$, contradicting our assumption (see Figure 8.26).

In the other case, when $u' \in S^{t_2}_{u_2}$, we look at the original graph H. Since both $v', v \in S''_v$, there exists a path P_v from v' to v using only cactus edges and vertices from S''_v. Similarly, since $u', u_2 \in S^{t_2}_{u_2}$, there exists a path $P_{u'}$ from u' to u_2 using only cactus edges and vertices from $S_{u_2}^{t_2}$. Hence, the region R bounded by $v'P_v v \cup vu_1 \cup u_2 P_{u'} u' \cup u'v'$ contains only the vertices from $S_v'' \cup S_{u_2}^{t_2}$ at its boundary and also contains the base edge vu_2 from the side outside of f. This implies that the super-face f_2 can only be embedded inside R and can only contain vertices from $S_v'' \cup S_{u_2}^{t_2}$ and hence for e_2 to go across the occupied components of t_2, the only possibility is to go across S_v'' and $S_{u_2}^{t_2}$, contradicting our assumption (see Figure 8.26). $\square$

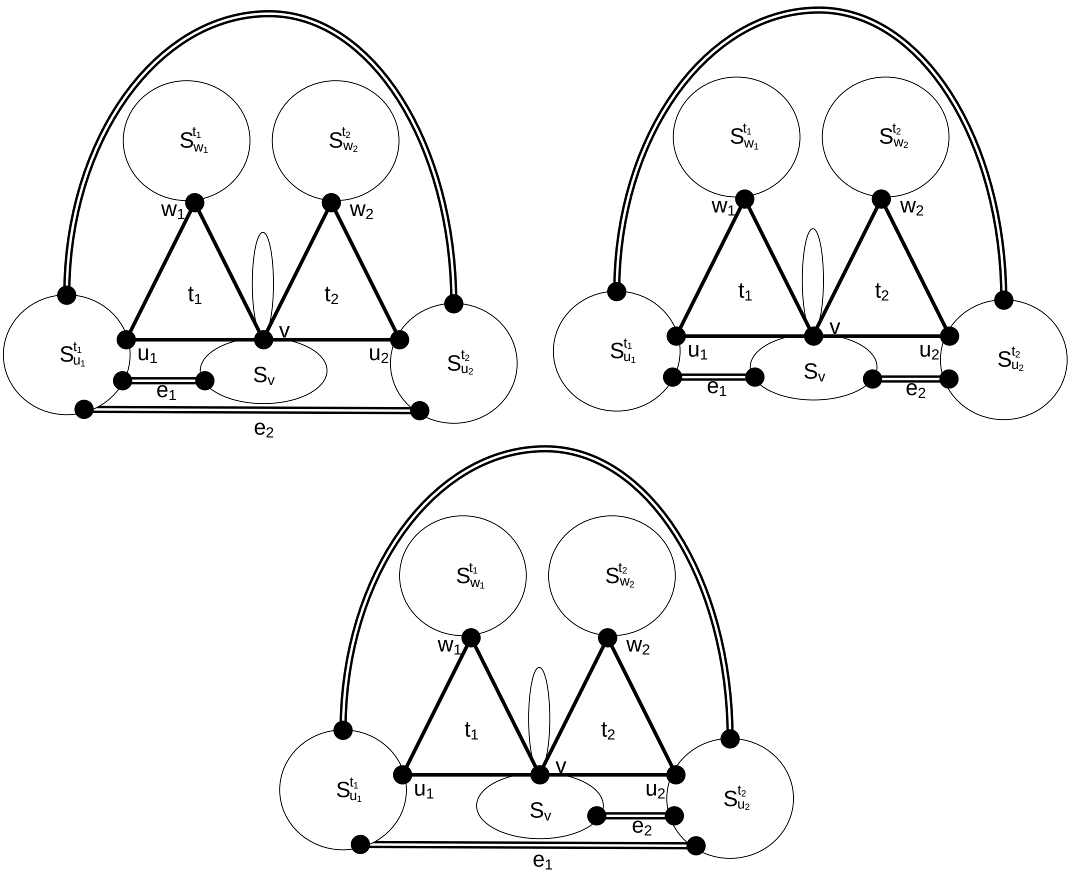


Figure 8.26: The possible embeddings when two adjacent heavy cactus triangles t_1 and t_2 are such that their base sides belong to two different super-faces $f_1, f_2 \in \mathcal{F}$. Here, e_1 and e_2 are the corresponding edges for t_1 and t_2 given by Lemma 8.52.

Using the presented structural properties, we are now able to prove the first property of Lemma 8.41.

Claim 8.57. *No type-*1 *heavy triangle is friends with any other heavy cactus triangle.*

Proof. We assume for contradiction that t_1 and t_2 are friends where t_2 is a type-1 triangle. We will argue that there exists an improving 2-swap, contradicting the fact that $\mathcal{C}$ is the optimal cactus. As t_1 and t_2 are adjacent, we use the notations defined above for the various components corresponding to two adjacent heavy triangles.

Let t' be the supported cross triangle of t_2 and let t_3 be the empty triangle formed by vertices $\{w_1, w_2, v\}$. Also let e_1, e_2 (possibly same) be the type-1 or type-2 edges belonging to the super-face f going across the occupied components of t_1 and t_2 respectively (exists by Lemma 8.52). Also, let e'_2 be the edge going across the occupied components of t_2 which belongs to the super-face f_2 containing the base side for t_2 (exists by Lemma 8.52). By Lemma 8.54, $e_2 \neq e'_2$.

Now there could be two cases based on the landing components for supported cross triangles t', t'_1. The second case will be further divided into sub-cases based on the way e_1 is embedded in ϕ_H.

- (e_1 is a type-1 edge and different landing components for supported cross triangles t'_1, t'): We modify our cactus by $\mathcal{C}' = (\mathcal{C}[S] \setminus (E(t_1) \cup E(t_2))) \cup E(t') \cup E(t'_1) \cup E(t_3)$ (see Figure 8.27). Note that t' will attach $S_2^{t_2}$ to S_v, t_3 will attach S_v with $S_{w_1}^{t_1}$ and $S_{w_2}^{t_2}$ and finally t'_1 will attach $S_{u_1}^{t_1}$ to this structure, hence $\mathcal{C}'$ will be a triangular cactus with one more cactus triangle, which contradicts the optimality of $\mathcal{C}$.

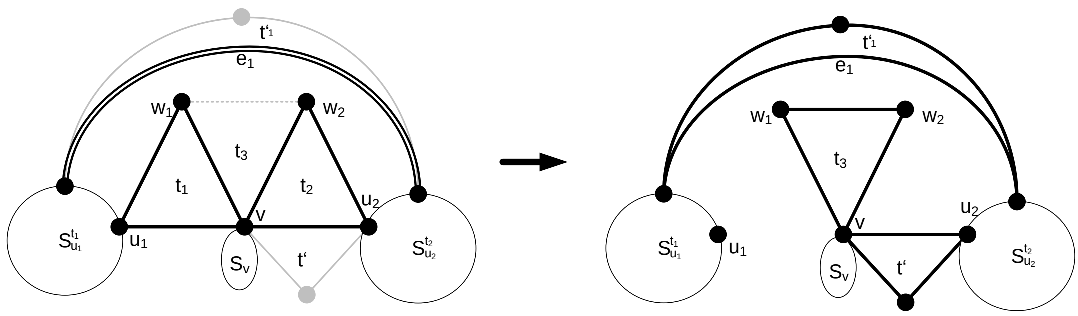


Figure 8.27: There exists a 2-swap, if e_1 given by Lemma 8.52 is a type-1 edge and the cross triangles supported by e_1 and t_2 have different landing components.

- (e_1 is a type-1 edge and t'_1, t' share a common landing component): Since e_1 is the unique edge belonging to f going across the occupied components of t_1 (see Lemma 8.52), there could be two sub-cases.

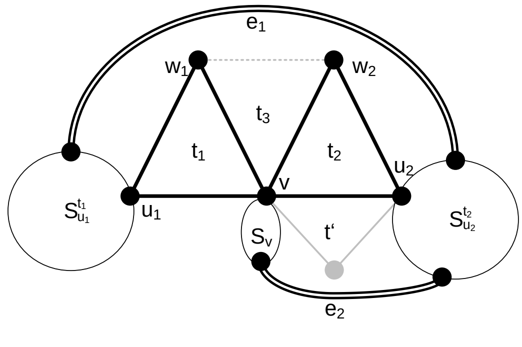


Figure 8.28: The case when e_1 given by Lemma 8.52 is a type-1 edge and the cross triangles supported by e_1 and t_2 have the same landing component.

– (e_1 goes across $S^{t_1}_{u_1}, S^{t_2}_{u_2}$): From Lemma 8.55 it implies that $e_1 = e_2 =: e$. Now if we focus on t_2, the base side of it must belong to a super-face f_2 such that e'_2 goes across its occupied components such that $f \neq f_2$ and $e'_2 \neq e$ (by Observation 8.53, Lemma 8.52 and 8.54). Also, by the uniqueness of the edge e'_2, the edge e cannot belong to f_2. But this implies that t' is embedded inside f_2 and t'_1 is embedded outside it, hence by Observation 8.3 t'_1, t' cannot share their landing components, contradiction.

– (e_1 goes across $S^{t_1}_{u_1}, S_v$): By Lemma 8.55, it implies $e_2 \neq e_1$ and both e_1, e_2 are incident to S_v. Now let $u'v' := e_2$ such that $u' \in S^{t_2}_{u_2}$ and $v' \in S_v$. Since both $v', v \in S_v$, there exists a path $P_{v'}$ from v' to v using only cactus edges and vertices from S_v. Similarly, since $u', u_2 \in S^{t_2}_{u_2}$, there exists a path $P_{u'}$ from u' to u_2 using only cactus edges and vertices from $S^{t_2}_{u_2}$. Hence, the region R bounded by $v'P_{v'}v \cup vu_2 \cup u_2P_{u'}u' \cup u'v'$ contains only the vertices from $S_v \cup S^{t_2}_{u_2}$ at its boundary and also contains the base edge vu_2 (see Figure 8.29). This implies that the super-face f_2 can only be embedded inside R and consecutively the triangle t' is embedded inside R. This implies that e_1 should be embedded outside R and consecutively t'_1 is embedded outside R, hence by Observation 8.3, t'_1 and t' cannot share their landing components, contradiction.

□

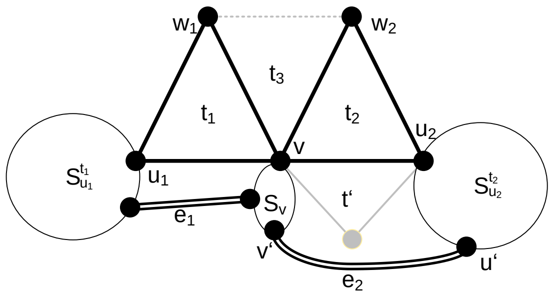


Figure 8.29: The setting before we reach a contradiction in the proof of Claim 8.57 for the case when e_1 given by Lemma 8.52 is a type-1 edge and the cross triangles supported by e_1 and t_2 have the same landing component.

We conclude the proof of Lemma 8.41 by showing that the second property holds.

Claim 8.58. *For any pair of type-0 triangles which are friends, their corresponding base sides belong to a common super-face in $\mathcal{F}$.*

Proof. For contradiction we assume that t_1 and t_2 are friends. Again t_1 and t_2 are adjacent, hence we use the notations defined above for the various components corresponding to two adjacent heavy triangles. Let t_3 be the triangle formed by vertices $\{w_1, w_2, v\}$. Also let e_1 be the unique type-1 or type-2 edge belonging to the super-face f_1 containing base side of t_1 going across occupied components of t_1 (exists by Lemma 8.52) and e_2 be the unique type-1 or type-2 edge belonging to the super-face f_2 containing base side of t_2 going across occupied components of t_2 (exists by Lemma 8.52). Let e'_1, e'_2 (possibly same) be the unique type-1 or type-2 edges belonging to the super-face f going across occupied components of t_1 and t_2 respectively (exists by Lemma 8.52 and the fact that f contains free sides for both t_1 and t_2). By Lemma 8.54 and 8.56, $e_1 \neq e_2$, $e_1 \neq e'_1$ and $e_2 \neq e'_2$.

Now we fix the cross triangles t'_1, t'_2, t''_1 each supported by e_1, e_2, e'_1 respectively, as follows. The idea here is to fix these supported cross triangles in such a way that their landing components are as different as possible. If e'_1 supports a cross triangle embedded inside f, then we fix t''_1 to be that triangle, otherwise t''_1 is any supported cross triangle of e'_1. If there exists a cross triangle supported by e_1 which does not share its landing component with t''_1 then we fix t'_1 to be that triangle, otherwise t'_1 is any supported cross triangle of e_1. Similarly, we choose the supported cross triangle t'_2 of e_2 such that it does not share its landing component with any of t''_1 or t'_1 (or both), otherwise t'_2 is any supported cross triangle of e_2.

By the way t'_1, t'_2, t''_1 are chosen, it ensures that all three of them can share a landing component if and only if all three e_1, e_2, e'_1 are type-1 edges (by Lemma 8.13). Now there could be three cases.

- (t'_1, t'_2 have different landing components): Since the base sides of t_1 and t_2 are in different super-faces, Lemma 8.56) implies that at least one of e_1, e_2 is incident to S_v (by renaming assume e_1). Hence, if the triangles t'_1, t'_2 do not share their landing components then we modify our cactus by $\mathcal{C}' = (\mathcal{C}[S] \setminus (E(t_1) \cup E(t_2))) \cup E(t'_1) \cup E(t'_2) \cup E(t_3)$ (See Figure 8.30). Note that t'_1 will attach $S^{t_1}_{u_1}$ to S_v, t_3 will attach S_v with $S^{t_1}_{w_1}$ and $S^{t_2}_{w_2}$ and finally t'_2 will attach $S^{t_2}_{u_2}$ to this structure, hence $\mathcal{C}'$ will be a triangular cactus with one more cactus triangle, which contradicts the optimality of $\mathcal{C}$.

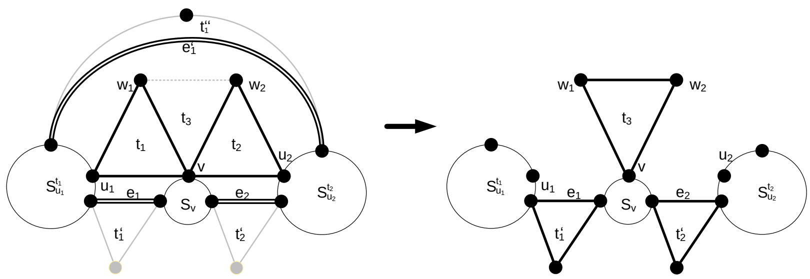


Figure 8.30: A improving 2-swap, if there exist two cross triangles t'_1, t'_2 supported by e_1, e_2 (as assumed to exist for the first case of the proof of Claim 8.58), respectively, such that their landing components are different.

- (t''_1 has a different landing component than the common landing component for t'_1, t'_2): In this case we know that t'_1, t'_2 share their landing components but the landing component for t''_1 is different. Again, since the base sides of t_1 and t_2 are in different super-faces, Lemma 8.56) implies that at least one of e_1, e_2 is incident to S_v. Now there are two sub-cases:

– (e_2 incident to S_v): In this case, we modify our cactus by $\mathcal{C}' = (\mathcal{C}[S] \setminus (E(t_1) \cup E(t_2))) \cup E(t_2') \cup E(t_1'') \cup E(t_3)$ (See Figure 8.31). Again t_2' will attach S_v with $S_{u_2}^{t_2}$, t_3 will attach S_v with $S_{w_1}^{t_1}$ and $S_{w_2}^{t_2}$ and finally t_1'' will attach $S_{u_1}^{t_1}$ to this structure, hence $\mathcal{C}'$ will be a triangular cactus with one more cactus triangle, which contradicts the optimality of $\mathcal{C}$.

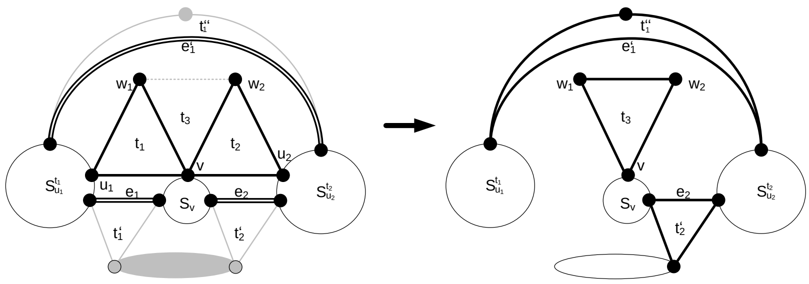


Figure 8.31: An improving 2-swap, if there exists two cross triangles t_2', t_1'' supported by e_2, e_1' (as assumed to exist for the second case of the proof of Claim 8.58), respectively, such that their landing components are different and e_2 incident to S_v.

– (Only e_1 incident to S_v): In this case e_2 goes across $S_{u_1}^{t_1}, S_{u_2}^{t_2}$.

  * (e_1' incident to S_v): The modification $\mathcal{C}' = (\mathcal{C}[S] \setminus (E(t_1) \cup E(t_2))) \cup E(t_2') \cup E(t_1'') \cup E(t_3)$ gives us the contradiction since t_1' will attach $S_{u_1}^{t_1}$ to S_v, t_3 will attach S_v with $S_{w_1}^{t_1}$ and $S_{w_2}^{t_2}$ and finally t_2' will attach $S_{u_2}^{t_2}$ to this structure, hence $\mathcal{C}'$ will be a triangular cactus with one more cactus triangle, which contradicts the optimality of $\mathcal{C}$.
  * (e_1' goes across $S_{u_1}^{t_1}, S_{u_2}^{t_2}$): The modification $\mathcal{C}' = (\mathcal{C}[S] \setminus (E(t_1) \cup E(t_2))) \cup E(t_1') \cup E(t_1'') \cup E(t_3)$ (see Figure 8.32) gives us the contradiction since t_1' will attach $S_{u_1}^{t_1}$ to S_v, t_3 will attach S_v with $S_{w_1}^{t_1}$ and $S_{w_2}^{t_2}$ and finally t_1'' will attach $S_{u_2}^{t_2}$ to this structure, hence C' will be a triangular cactus with one more cactus triangle, which contradicts the optimality of $\mathcal{C}$.

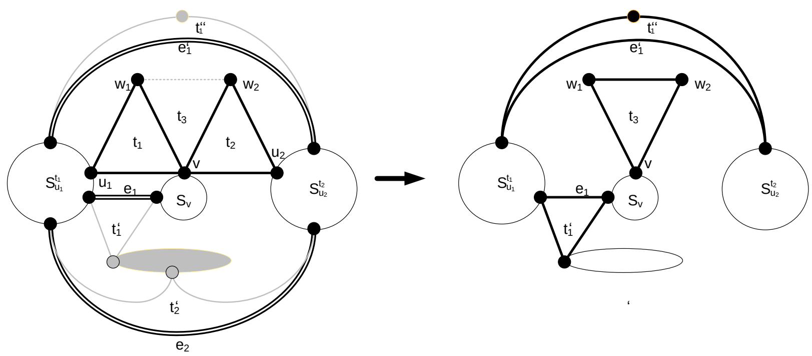


Figure 8.32: An improving 2-swap, if there exists two cross triangles t_2', t_1'' supported by e_2, e_1' (as assumed to exist for the second case of the proof of Claim 8.58), respectively, such that their landing components are different and e_1' goes across $S_{u_1}^{t_1}, S_{u_2}^{t_2}$.

- (All three triangles t_1', t_2', t_1'' share their landing components): In this case, all three e_1, e_2, e_1' are type-1 edges. Also by Lemma 8.56, at least one of e_1, e_2 will be incident to S_v. And since $S_{w_1}^{t_1}, S_{w_1}^{t_2}$ are free components, none of the three edges e_1, e_2, e_1' can be incident to $S_{w_1}^{t_1}, S_{w_1}^{t_2}$. Based on these facts, there could be two sub-cases:

  - (Exactly one of e_1 or e_2 is incident to S_v): We will argue that this case cannot occur, by showing that there is no way for t_1'' to share the same landing component with t_1', t_2'. Since t_1 and t_2 are friends, all the vertices of S_v (except v) are embedded outside t_3. This also implies that there is a trail P starting from vertex u_1, using all the cactus/type-1/type-2 edges on the outer-face for $H[S_v]$ and finally reaching u_2, such that the only repeated vertex in the trail is v. Since exactly one of e_1 or e_2 is incident to S_v, this implies that the other one goes across $S_{u_1}^{t_1}, S_{u_2}^{t_2}$ (say $u'v'$) such that $u' \in S_{u_1}^{t_1}$ and $v' \in S_{u_2}^{t_2}$. This means that there exists a circuit C comprising of only cactus/type-1/type-2 edges formed by concatenating the trail P, the path $P_{u'}$ between u' and u_1 using cactus edges/vertices only from $S_{u_1}^{t_1}$, the path $P_{v'}$ between v' and u_2 using cactus edges/vertices only from $S_{u_2}^{t_2}$ and the type-1 or type-2 edge $u'v'$. It is easy to see that this circuit partitions the plane into two regions, say R_1, R_2, such that all the vertices of S_v are embedded inside R_1 as a hole and the free sides for t_1 and t_2 are embedded in R_2 such that the only vertex from S_v on the boundary for these regions is v. Also, the presence of the edge w_1w_2 does not allow the vertex v to be a part of any type-1 or type-2 edge embedded inside R_2. This implies that the edge out of e_1, e_2 which is incident to S_v will be embedded inside R_1 and e_1' will be embedded outside R_2, which contradicts the fact that all three supported cross triangles t_1', t_2', t_1'' share their landing component.

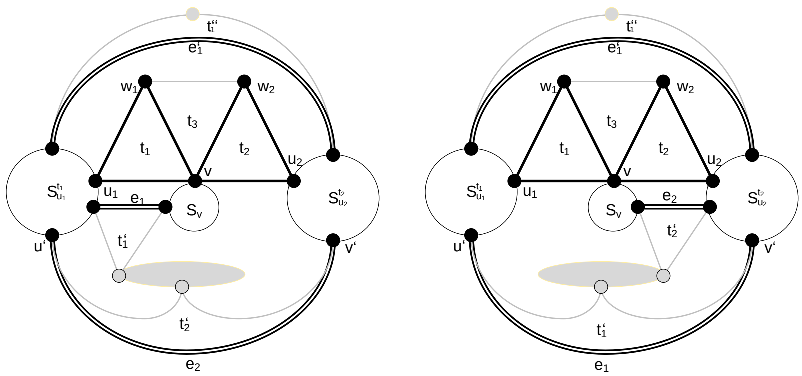


Figure 8.33: The setting before we reach a contradiction in the proof of Claim 8.58, if all three edges e_1, e_2, e_1' are type-1 and the landing component for the respective supported cross triangles t_1', t_2', t_1'' is the same.

  - (Both e_1 and e_2 are incident to S_v): Now we focus on t_1, which is a heavy type-0 triangle and look at the type-1 or type-2 edges going across t_1's occupied components. The two type-1 edges e_1 and e_1' are surely going across the occupied components of t_1. By Proposition 8.5, t_1 should have at least one more such type-1 or type-2 edge (say $e_1'' := u'v'$). Now let $u' \in S_{u_1}^{t_1}$ and $v' \in S_{v_1}^{t_1}$. This means that there is a path $P_{u'}$ from u' to u_1 in $\mathcal{C}[S]$ and another path $P_{v'}$ from v' to $v_1 = v$ in $\mathcal{C}[S]$ such that the cycle $C_1 := u'P_{u'}u_1 \cup u_1v \cup vP_{v'}v' \cup u'v'$ is made of only cactus/type-1/type-2 edges and cactus vertices such that it divided the plane into two regions such that one region contains the base side of t_1 and another contains the free side for t_1. Since, e_1, e_1' see the base and free sides for t_1 respectively, hence they have to be embedded in the different region bounded by C_1. Hence, the cross triangles t_1', t_1'' supported by e_1, e_1' cannot share their landing components, which is a contradiction.

□

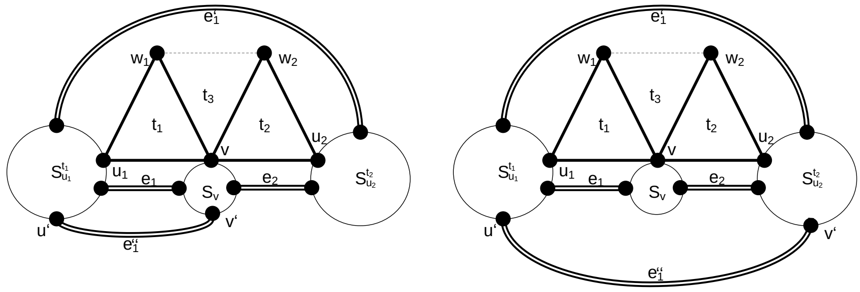


Figure 8.34: The setting before we reach a contradiction in the proof of Claim 8.58, if the edges e_1, e_2, e_1' are type-1 and the landing component for the respective supported cross triangles t_1', t_2', t_1'' is the same. Also, both e_1 and e_2 are incident to S_v.

8.5.5 Proof of Lemma 8.47

Below, we analyze the contribution from non outer-faces.

Coordinates	Value
$\vec{\chi}[1]$	$\lvert\mathcal{F}[1,0,0] \setminus \mathcal{F}_{fri}[1,0,0]\rvert$
$\vec{\chi}[2]$	$\lvert\mathcal{F}_{fri}[1,0,0]\rvert$
$\vec{\chi}[3]$	$\lvert\mathcal{F}[1,0,\geq 1]\rvert$
$\vec{\chi}[4]$	$\lvert\mathcal{F}[1,1,0] \setminus \mathcal{F}_{fri}[1,1,0]\rvert$
$\vec{\chi}[5]$	$\lvert\mathcal{F}_{fri}[1,1,0]\rvert$
$\vec{\chi}[6]$	$\lvert\mathcal{F}[1,1,\geq 1]\rvert$
$\vec{\chi}[7]$	$\lvert\mathcal{F}[2,0,0] \setminus \mathcal{F}_{fri}[2,0,0]\rvert$
$\vec{\chi}[8]$	$\lvert\mathcal{F}_{fri}[2,0,0]\rvert$
$\vec{\chi}[9]$	$\lvert\mathcal{F}[2,0,\geq 1]\rvert$
$\vec{\chi}[10]$	$\lvert\mathcal{F}[2,1,\bullet]\rvert$
$\vec{\chi}[11]$	$\lvert\mathcal{F}[2,2,\bullet]\rvert$
$\vec{\chi}[12]$	$\lvert\mathcal{F}[\geq 3,\bullet,\bullet]\rvert$

Table 8.1: Definition of characteristic vector of $\mathcal{F}$

This is simply an algebraic manipulation. First, we write

$$\overrightarrow{gain} \cdot \vec{\chi} \geq \frac{9}{2}(\mathbb{1}^T \vec{\chi}) - (0, 2, 2, \frac{1}{2}, \frac{5}{2}, \frac{5}{2}, \frac{3}{2}, \frac{5}{2}, \frac{5}{2}, 2, \frac{5}{2}, 3)^T \vec{\chi}.$$

We will gradually decompose the vector

$$(0, 2, 2, \frac{1}{2}, \frac{5}{2}, \frac{5}{2}, \frac{3}{2}, \frac{5}{2}, \frac{5}{2}, 2, \frac{5}{2}, 3)^T \vec{\chi}.$$

into several meaningful terms that we could upper bound. First, we focus on the coordinates that correspond to the η_{fri} (highlighted in blue):

$$(0, 2, 2, \frac{1}{2}, \frac{5}{2}, \frac{5}{2}, \frac{3}{2}, \frac{5}{2}, \frac{5}{2}, 2, \frac{5}{2}, 3)^T \vec{\chi} = \boxed{2\eta_{fri}} + (0, 0, 2, \frac{1}{2}, \frac{1}{2}, \frac{5}{2}, \frac{3}{2}, \frac{1}{2}, \frac{5}{2}, 2, \frac{5}{2}, 3)^T \vec{\chi},$$

where we simply applied the fact that $\eta_{fri}[1,0,0] + \eta_{fri}[1,1,0] + \eta_{fri}[2,0,0] = \eta_{fri}$. Next, we focus on the components of $\eta[2, \bullet, \bullet]$ and $\eta[3, \bullet, \bullet]$ (shown in red).

$$\begin{aligned}(0, 0, 2, \frac{1}{2}, \frac{1}{2}, \frac{5}{2}, \frac{3}{2}, \frac{1}{2}, \frac{5}{2}, 2, \frac{5}{2}, 3)^T \vec{\chi} &\leq \boxed{\tfrac{3}{2}(p_1 + |\mathcal{F}| - 2)} \\ &\quad + (0, 0, 2, \frac{1}{2}, \frac{1}{2}, \frac{5}{2}, 0, -1, 1, \frac{1}{2}, 1, 0)^T \vec{\chi},\end{aligned}$$

where we applied the upper bound from Lemma 8.46 (first bound). We further extract the "components" of $\eta[1,1,0]$, $\eta[2,1,\bullet]$ and $\eta[2,2,\bullet]$:

$$\begin{aligned}(0, 0, 2, \frac{1}{2}, \frac{1}{2}, \frac{5}{2}, 0, -1, 1, \frac{1}{2}, 1, 0)^T \vec{\chi} &= \frac{1}{2}(\eta[1,1,0] + \eta[2,1,\bullet] + 2\eta[2,2,\bullet]) \\ &\quad + (0, 0, 2, 0, 0, 2, 0, -1, 1, 0, 0, 0)^T \vec{\chi} \\ &\leq \boxed{\tfrac{1}{2} a_1} + (0, 0, 2, 0, 0, 2, 0, -1, 1, 0, 0, 0)^T \vec{\chi}\end{aligned}$$

the inequality was obtained by applying Lemma 8.46 (second bound). Now, we extract the components of $\eta[1,1,\geq 1]$, $\eta[2,0,\geq 1]$ and $\eta[1,0,\geq 1]$ (the 3rd, 6th, and 9th coordinates respectively).

$$\begin{aligned}(0, 0, 2, 0, 0, 2, 0, -1, 1, 0, 0, 0)^T \vec{\chi} &= 2(\eta[1,0,\geq 1] + \eta[1,1,\geq 1] + \eta[2,0,\geq 1]) \\ &\quad + (0, 0, 0, 0, 0, 0, 0, -1, -1, 0, 0, 0)^T \vec{\chi} \\ &\leq 2(p_0 - \eta_{fri}) + (0, 0, 0, 0, 0, 0, 0, -1, -1, 0, 0, 0)^T \vec{\chi} \\ &\leq \boxed{2(p_0 - \eta_{fri})}.\end{aligned}$$

Here we applied the third bound of Lemma 8.46, and the fact that all coordinates of vector $\vec{\chi}$ are non-negative. Finally, by summing over all terms in the boxes, we get the upper bound of

$$2\eta_{fri} + \frac{3}{2}(p_1 + |\mathcal{F}| - 2) + \frac{1}{2}a_1 + 2(p_0 - \eta_{fri}) = 2p - \frac{1}{2}p_1 + 2a_1 + \frac{3}{2}a_2 - \frac{3}{2}.$$

Now, since $\mathbb{1}^T\vec{\chi} = a_1 + a_2$ and $gain(f_0) \geq \phi(S) - 1$, we have that

$$\sum_{f \in \mathcal{F}} gain(f) \geq \frac{9}{2}(a_1 + a_2) - (2p - \frac{1}{2}p_1 + 2a_1 + \frac{3}{2}a_2 - \frac{3}{2}) + \phi(S) - 1.$$

Hence, $-(\sum_{f \in \mathcal{F}} gain(f)) \leq -\phi(S) + (2p - \frac{1}{2}p_1 - 3a_2 - \frac{5}{2}a_1 - \frac{1}{2})$.

We substitute the bound from the lemma into Equation 8.2, we would get:

$$q \leq (4p + \frac{1}{2}p_1 + \frac{5}{2}a_1 + 3a_2) - \phi(S) + (2p - \frac{1}{2}p_1 - 3a_2 - \frac{5}{2}a_1 - \frac{1}{2}).$$

This gives $q \leq 6p - \phi(S) - \frac{1}{2}$ as desired.

CHAPTER 9

Conclusion

The new approach for solving MPS by concentrating on triangles, introduced in this thesis, allowed us immediately to improve over the previously best-known greedy algorithms for MPS. In addition, the results shown in this part imply that a natural local search algorithm gives a $(\frac{4}{9}+\varepsilon)$-approximation for MPS and a $\frac{1}{6}+\varepsilon$ approximation for MPT. To be more precise, when given any graph G, we follow the t-swap local search strategy for $t = O(1/\varepsilon)$: Start from any cactus subgraph H. Try to improve it by removing t triangles and adding $(t+1)$ triangles in a way that ensures that the graph remains a cactus subgraph. A local optimal solution will then always be a $(\frac{4}{9}+\varepsilon)$ approximation for MPS and a $(\frac{1}{6}+\varepsilon)$ approximation for MPT.

Knowing this fact, there is an obvious candidate algorithm for improving over the long-standing best approximation factor for MPS. We call a graph H a diamond-cactus if every block in H is either a diamond or a triangle. Start from any diamond-cactus subgraph H of G and then try to improve it by removing t triangles from H and adding $(t+1)$ triangles, maintaining the fact that H is a diamond-cactus subgraph. We conjecture that this algorithm gives a better than $\frac{4}{9}$-approximation for MPS, but we suspect that the analysis will require substantially new ideas.

Another interesting direction is to see whether there is a general principle that captures a denser planar structure than cactus subgraphs by going above matroid parity in the hierarchy of efficiently computable problems. For instance, are diamond-cactus subgraphs captured by matroid parity? Or can it be formulated as an even more abstract structure than matroids (e.g. commutative rank [6]) that can still be computed efficiently? We believe that studying this direction will lead to a better understanding of algebraic techniques for finding dense planar structures.

Finally, the absence of LP-based techniques in this problem domain seems rather unfortunate. There have been some experimental studies recently [41, 20, 21], but the theoretical understanding of what can be proven formally in the context of power of relaxation is certainly lacking. Is there a convex relaxation that allows us to find a relatively dense planar subgraph (e.g. $(3-\varepsilon)$-approximation for MPS using LP-based techniques)?

Bibliography

[1] T. Asano, S. Kikuchi, and N. Saito. A linear algorithm for finding Hamiltonian cycles in 4-connected maximal planar graphs. *Discrete Applied Mathematics*, 7(1):1–15, 1984.

[2] D. Barnette. Trees in polyhedral graphs. *Canadian Journal of Mathematics*, 18:731–736, 1966.

[3] C. Berge. La theorie des graphes. *Paris, France*, 1958.

[4] T. Biedl. Trees and co-trees with bounded degrees in planar 3-connected graphs. In *14th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT'14)*, pages 62–73, 2014.

[5] T. Biedl and P. Kindermann. Finding Tutte Paths in Linear Time. In C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[6] M. Bläser, G. Jindal, and A. Pandey. Greedy strikes again: A deterministic PTAS for commutative rank of matrix spaces. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 33:1–33:16, 2017.

[7] G. Brinkmann and C. T. Zamfirescu. A Strengthening of a Theorem of Tutte on Hamiltonicity of Polyhedra. *ArXiv e-prints*, 2016.

[8] R. Brunet, M. N. Ellingham, Z. C. Gao, A. Metzlar, and R. B. Richter. Spanning planar subgraphs of graphs in the torus and Klein bottle. *Journal of Combinatorial Theory, Series B*, 65(1):7–22, 1995.

[9] J. Cai, X. Han, and R. E. Tarjan. An $O(m \log n)$-time algorithm for the maximal planar subgraph problem. *SIAM Journal on Computing*, 22(6):1142–1162, 1993.

[10] G. Călinescu, C. G. Fernandes, U. Finkler, and H. Karloff. A better approximation algorithm for finding planar subgraphs. *Journal of Algorithms*, 27(2):269–302, 1998.

[11] G. Călinescu, C. G. Fernandes, H. J. Karloff, and A. Zelikovsky. A new approximation algorithm for finding heavy planar subgraphs. *Algorithmica*, 36(2):179–205, 2003.

[12] G. Călinescu, C. G. Fernandes, H. Kaul, and A. Zelikovsky. Maximum series-parallel subgraph. *Algorithmica*, 63(1-2):137–157, 2012.

[13] P. Chalermsook, M. Cygan, G. Kortsarz, B. Laekhanukit, P. Manurangsi, D. Nanongkai, and L. Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 743–754. IEEE, 2017.

[14] P. Chalermsook and A. Schmid. Finding triangles for maximum planar subgraphs. In *WALCOM: Algorithms and Computation, 11th International Conference and Workshops, (WALCOM'17), Proceedings.*, pages 373–384, 2017.

[15] P. Chalermsook, A. Schmid, and S. Uniyal. A Tight Extremal Bound on the Lovász Cactus Number in Planar Graphs. In R. Niedermeier and C. Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[16] H. Y. Cheung, L. C. Lau, and K. M. Leung. Algebraic algorithms for linear matroid parity problems. *ACM Transactions on Algorithms (TALG)*, 10(3):10, 2014.

[17] N. Chiba and T. Nishizeki. A theorem on paths in planar graphs. *Journal of Graph Theory*, 10(4):449–450, 1986.

[18] N. Chiba and T. Nishizeki. The Hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *Journal of Algorithms*, 10(2):187–211, 1989.

[19] T. Chiba, I. Nishioka, and I. Shirakawa. An algorithm of maximal planarization of graphs. In *Proc. IEEE Symposium on Circuits and Systems, 1979*, pages 649–652.

[20] M. Chimani, I. Hedtke, and T. Wiedera. Exact algorithms for the maximum planar subgraph problem: New models and experiments. In *17th International Symposium on Experimental Algorithms, SEA 2018, June 27-29, 2018, L'Aquila, Italy*, pages 22:1–22:15, 2018.

[21] M. Chimani and T. Wiedera. Cycles to the Rescue! Novel Constraints to Compute Maximum Planar Subgraphs Fast. In Y. Azar, H. Bast, and G. Herman, editors, *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[22] R. Cimikowski and D. Coppersmith. The sizes of maximal planar, outerplanar, and bipartite planar subgraphs. *Discrete Math.*, 149(1-3):303–309, Feb. 1996.

[23] R. Diestel. *Graph Theory.* Springer, fourth edition, 2010.

[24] I. Fabrici, J. Harant, and S. Jendrol. On longest cycles in essentially 4-connected planar graphs. *Discussiones Mathematicae Graph Theory*, 36:565–575, 2016.

[25] I. Fabrici, J. Harant, S. Mohr, and J. M. Schmidt. Even longer cycles in essentially 4-connected planar graphs. *arXiv preprint arXiv:1806.09413*, 2018.

[26] I. Fabrici, J. Harant, S. Mohr, and J. M. Schmidt. Longer cycles in essentially 4-connected planar graphs. *Discussiones Mathematicae Graph Theory*, to appear.

[27] L. Faria, C. M. H. De Figueiredo, and C. F. Mendonça. On the complexity of the approximation of nonplanarity parameters for cubic graphs. *Discrete applied mathematics*, 141(1):119–134, 2004.

[28] H. N. Gabow and M. Stallmann. An augmenting path algorithm for linear matroid parity. *Combinatorica*, 6(2):123–150, 1986.

[29] Z. Gao and R. B. Richter. 2-Walks in circuit graphs. *Journal of Combinatorial Theory, Series B*, 62(2):259–267, 1994.

[30] Z. Gao, R. B. Richter, and X. Yu. 2-Walks in 3-connected planar graphs. *Australasian Journal of Combinatorics*, 11:117–122, 1995.

[31] Z. Gao, R. B. Richter, and X. Yu. Erratum to: 2-Walks in 3-connected planar graphs. *Australasian Journal of Combinatorics*, 36:315–316, 2006.

[32] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5(4):704–714, 1976.

[33] D. Gouyou-Beauchamps. The Hamiltonian circuit problem is polynomial for 4-connected planar graphs. *SIAM Journal on Computing*, 11(3):529–539, 1982.

[34] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. In *Graph Algorithms And Applications 2*, pages 3–18. World Scientific, 2004.

[35] J. Harant and S. Senitsch. A generalization of Tutte's theorem on Hamiltonian cycles in planar graphs. *Discrete Mathematics*, 309(15):4949–4951, 2009.

[36] F. Harary and G. Prins. The block-cutpoint-tree of a graph. *Publ. Math. Debrecen*, 13:103–107, 1966.

[37] J. Håstad. Clique is hard to approximate withinn 1- ε. *Acta Mathematica*, 182(1):105–142, 1999.

[38] B. Jackson and N. C. Wormald. k-Walks of graphs. *Australasian Journal of Combinatorics*, 2:135–146, 1990.

[39] B. Jackson and N. C. Wormald. Longest cycles in 3-connected planar graphs. *Journal of Combinatorial Theory, Series B*, 54(2):291–321, 1992.

[40] B. Jackson and X. Yu. Hamilton cycles in plane triangulations. *Journal of Graph Theory*, 41(2):138–150, 2002.

[41] M. Jünger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16(1):33–59, Jul 1996.

[42] K. Kawarabayashi and K. Ozeki. 4-connected projective-planar graphs are Hamiltonian-connected. *Journal of Combinatorial Theory, Series B*, 112:36–69, 2015.

[43] S. Khot and A. K. Ponnuswami. Better inapproximability results for maxclique, chromatic number and min-3lin-deletion. In *International Colloquium on Automata, Languages, and Programming*, pages 226–237. Springer, 2006.

[44] M. S. Krishnamoorthy. An NP-hard problem in bipartite graphs. *ACM SIGACT News*, 7(1):26–26, 1975.

[45] D. Kühn, D. Osthus, and A. Taraz. Large planar subgraphs in dense graphs. *Journal of Combinatorial Theory, Series B*, 95(2):263–282, 2005.

[46] J. Lee, M. Sviridenko, and J. Vondrák. Matroid matching: the power of local search. *SIAM Journal on Computing*, 42(1):357–379, 2013.

[47] T. Leighton and A. Moitra. Some results on greedy embeddings in metric spaces. *Discrete & Computational Geometry*, 44(3):686–705, Oct 2010.

[48] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

[49] P. Liu and R. Geldmacher. On the deletion of nonplanar edges of a graph. In *Proc. 10th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 727–738, 1977.

[50] L. Lovász. Matroid matching and some applications. *Journal of Combinatorial Theory, Series B*, 28(2):208–236, 1980.

[51] L. Lovász and M. D. Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[52] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *International Colloquium on Automata, Languages, and Programming*, pages 40–51. Springer, 1993.

[53] A. Nakamoto, Y. Oda, and K. Ota. 3-trees with few vertices of degree 3 in circuit graphs. *Discrete Mathematics*, 309(4):666 – 672, 2009.

[54] J. B. Orlin. A fast, simpler algorithm for the matroid parity problem. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 240–258. Springer, 2008.

[55] K. Ozeki. A shorter proof of Thomassen's theorem on Tutte paths in plane graphs. *SUT Journal of Mathematics*, 50:417–425, 2014.

[56] K. Ozeki and P. Vrána. 2-edge-Hamiltonian-connectedness of 4-connected plane graphs. *European Journal of Combinatorics*, 35:432–448, 2014.

[57] T. Poranen. Two new approximation algorithms for the maximum planar subgraph problem. *Acta Cybern.*, 18(3):503–527, 2008.

[58] D. P. Sanders. On Hamilton cycles in certain planar graphs. *Journal of Graph Theory*, 21(1):43–50, 1996.

[59] D. P. Sanders. On paths in planar graphs. *Journal of Graph Theory*, 24(4):341–345, 1997.

[60] A. Schmid. 2-walks in 3-connected planar graphs are polynomial time computable. 2014.

[61] A. Schmid and J. M. Schmidt. Computing 2-walks in polynomial time. In *Proceedings of the 32nd Symposium on Theoretical Aspects of Computer Science (STACS'15)*, pages 676–688, 2015.

[62] A. Schmid and J. M. Schmidt. Computing 2-walks in polynomial time. *ACM Trans. Algorithms*, 14(2):22:1–22:18, Apr. 2018.

[63] A. Schmid and J. M. Schmidt. Computing Tutte Paths. In I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 98:1–98:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[64] J. M. Schmidt. A simple test on 2-vertex- and 2-edge-connectivity. *Information Processing Letters*, 113(7):241–244, 2013.

[65] W.-B. Strothmann. *Bounded degree spanning trees.* PhD thesis, FB Mathematik/Informatik und Heinz Nixdorf Institut, Universität-Gesamthochschule Paderborn, 1997.

[66] Z. Szigeti. On a min-max theorem of cacti. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 84–95. Springer, 1998.

[67] R. Thomas and X. Yu. 4-connected projective-planar graphs are Hamiltonian. *Journal of Combinatorial Theory, Series B*, 62(1):114–132, 1994.

[68] R. Thomas and X. Yu. Five-connected toroidal graphs are Hamiltonian. *Journal of Combinatorial Theory, Series B*, 69(1):79–96, 1997.

[69] R. Thomas, X. Yu, and W. Zang. Hamilton paths in toroidal graphs. *Journal of Combinatorial Theory, Series B*, 94(2):214–236, 2005.

[70] C. Thomassen. A theorem on paths in planar graphs. *Journal of Graph Theory*, 7(2):169–176, 1983.

[71] W. T. Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 1(2):107–111, 1947.

[72] W. T. Tutte. A theorem on planar graphs. *Transactions of the American Mathematical Society*, 82:99–116, 1956.

[73] W. T. Tutte. Bridges and Hamiltonian circuits in planar graphs. *Aequationes Mathematicae*, 15(1):1–33, 1977.

[74] H. Whitney. A theorem on graphs. *Annals of Mathematics*, 32(2):378–390, 1931.

[75] H. Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 34(1):339–362, 1932.

[76] X. Yu. Disjoint paths, planarizing cycles, and spanning walks. *Transactions of the American Mathematical Society*, 349(4):1333–1358, 1997.