A Computational Approach for Finding 6-List-Critical Graphs on the Torus

Félix Moreno Peñarrubia

Treball de Final de Grau

Grau en Matemàtiques, FME-UPC
Grau en Enginyeria Informàtica, FIB-UPC

May 2023

Director: Zdeněk Dvořák *Tutor: Oriol Serra Albó*

Todo list

Abstract

Coloring graphs embedded on surfaces is an old and well-studied area of graph theory. Thomassen proved that there are finitely many 6-critical graphs on any fixed surface and provided the explicit set of 6-critical graphs on the torus. Later, Postle proved that there are finitely many 6-list-critical graphs on any fixed surface. With the goal of finding the set of 6-list-critical graphs on the torus, we develop and implement algorithmic techniques for computer search of critical graphs in different list-coloring settings.

Keywords: graphs on surfaces, list coloring, graph algorithms.

MSC2020 codes: 05C10, 05C15, 68R10.

Resum

La coloració de grafs dibuixats a superfícies és un àrea antiga i molt estudiada de la teoria de grafs. Thomassen va demostrar que hi ha un nombre finit de grafs 6-crítics a qualsevol superfície fixa i va proporcionar el conjunt explícit dels grafs 6-crítics al torus. Després, Postle va demostrar que hi ha un nombre finit de grafs 6-llista-crítics a qualsevol superfície fixa. Amb l'objectiu de trobar el conjunt de grafs 6-llista-crítics al torus, desenvolupem i implementem tècniques algorítmiques per la cerca per ordinador de grafs crítics en diferents situacions de coloració per llistes.

Paraules clau: grafs en superfícies, coloració amb llistes, algorismes sobre grafs.

Codis MSC2020: 05C10, 05C15, 68R10.

Resumen

La coloración de grafos dibujados en superficies es un área antigua y muy estudiada de la teoría de grafos. Thomassen demostró que hay un número finito de grafos 6-críticos en cualquier superficie fija y proporcionó el conjunto explícito de los grafos 6-críticos en el toro. Después, Postle demostró que hay un número finito de grafos 6-lista-críticos en cualquier superficie fija. Con el objetivo de encontrar el conjunto de grafos 6-lista-críticos en el toro, desarrollamos e implementamos técnicas algorítmicas para la búsqueda por ordenador de grafos críticos en diferentes situaciones de coloración por listas.

Palabras clave: grafos en superficies, coloración con listas, algoritmos sobre grafos.

Códigos MSC2020: 05C10, 05C15, 68R10.

Contents

fix function restrictions

Chapter 1

Introduction

In this chapter, we lay out the basic definitions of graph theoretical and topological concepts used in this thesis, as well as the background results which contextualize our research and an outline of the results we have obtained.

1.1 Graphs and Surfaces

1.1.1 Graph Theory Terminology

A *graph* is a pair $(V(G), E(G))$ consisting of a set $V(G)$ and a set $E(G)$ of two-element subsets of $V(G)$.

(connectivity, complete graphs, etc)

1.1.2 Surfaces

A *surface* is a...

(surface classification theorem, orientability, etc)

1.1.3 Embedding Graphs in Surfaces

A graph is embedded in a surface if...

(non-contractible cicles, planarity, dual graph, etc)

references for all this

1.2 Graph Coloring

Problems related to *coloring* are a fundamental part of graph theory. Although there are many variants, the original one and the most important is *vertex coloring*.

Definition 1.2.1. A *vertex coloring* of a graph G is a function $f : V(G) \to \mathbb{N}$. The vertex coloring is said to be *proper* if $\forall uv \in E(G), f(u) \neq f(v)$.

We think of this as assigning one color to each vertex of the graph, so that adjacent vertices are assigned different colors. This interpretation comes from the origin of the problem in *map coloring*, in which we have to color a political map assigning colors to countries so that neighboring countries are assigned different colors in order to distinguish them. A quantity of interest is the number of colors required for a proper coloring of the graph:

Definition 1.2.2. A vertex coloring f is said to be a *k-coloring* if $|\Im f| = k$. A graph G is said to be *k-colorable* if it admits a proper k-coloring. The *chromatic number* $\chi(G)$ of a graph G is the minimum k such that G is k-colorable.

In the map coloring context, we study vertex coloring for *planar* graphs. The following remarkable result was the origin of this area of mathematics:

Theorem 1.2.3 (Four color theorem)**.** *For all planar graphs G, $\chi(G) \leq 4$.*

This theorem, originally conjectured in TODO, was proven by Appel and Haken in TODO. Their proof achieved some notoriety due to use of computers to process a lengthy case analysis.

4ct dates and references

A natural generalization of the above problem is to study the chromatic number of graphs embedded in surfaces other than the plane. The following result, due to Heawood in 1890 [4], generalizes the four color theorem to surfaces other than the plane:

Theorem 1.2.4 (Heawood)**.** *Let Σ be a surface with Euler genus $g(\Sigma) \geq 1$. Any graph embedded in Σ can be colored with*

$$H(\Sigma) = \left\lfloor \frac{7 + \sqrt{1 + 24g(\Sigma)}}{2} \right\rfloor$$

colors.

We call $H(\Sigma)$ the *Heawood number* of the surface.

Interestingly, this theorem has a much simpler proof than the four color theorem. For its proof, we need the following concept:

Definition 1.2.5. A graph is *k-degenerate* if every subgraph $H \subseteq G$ has $\delta(H) \leq k$.

Proposition 1.2.6. *If a graph G is k-degenerate, then it is $(k+1)$-colorable.*

Proof. For each fixed $k \geq 0$, we do induction on the size of $|V(G)|$. For $|V(G)| = 1$, it is clearly true. Assuming it is true for all k-degenerate graphs G' with $|V(G')| < n$, let G be a k-degenerate graph with $|V(G)| = n$. Let v be a vertex of G with degree at most v and consider the graph $G\ \{v\}$. It is also a k-degenerate graph since it is a subgraph of G, so it is $(k+1)$-colorable. We can then extend the coloring to G by choosing a color not in any of its at most k neighbors for v. ∎

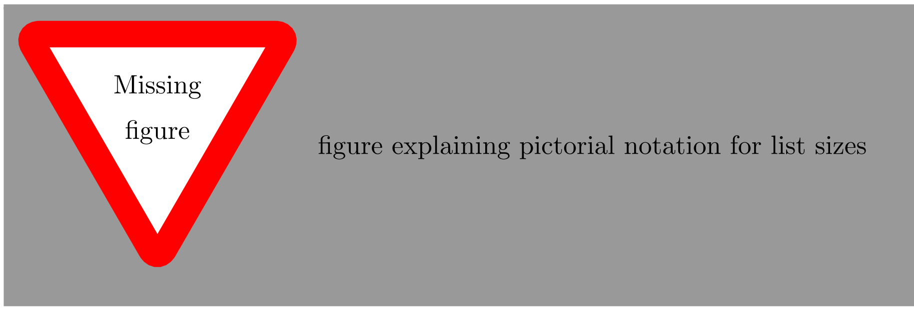
Proof of Heawood's theorem

Ringel-Youngs

Critical graphs, Hájos construction

Results in graphs on surfaces,

1.3 List Coloring



Thomassen's theorem introduction

Definition of List Coloring

[10]

Theorem 1.3.1 (Thomassen's theorem)**.** *For all planar graphs G, $\chi_\ell(G) \leq 5$.*

Thomassen actually proved a stronger theorem:

Theorem 1.3.2 (Thomassen's stronger theorem)**.** *Let G be a plane (embedded) graph with outer walk C, and let L be a list assignment satisfying:*

- $|L(v)| \geq 5$ *for all internal vertices.*
- $|L(v)| \geq 3$ *for all* $v \in V(C) \setminus \{x, y\}$ *where* x, y *are a pair of adjacent vertices.*
- $|L(x)| = |L(y)| = 1$, $L(x) \neq L(y)$.

Then G is L-colorable.

Proof. Suppose we have a counterexample with minimal $|V(G)|$. It is clear that for $|V(G)| \leq 3$ the theorem is true, so we assume $|V(G)| \geq 4$.

First we prove that G is 2-connected. Assume it is not. Then, we have two subgraphs $G_1, G_2 \subset G$ with $G_1 \cup G_2 = G$ and $G_1 \cap G_2 = \{v\}$, with v a cutvertex. Assume, without loss of generality, that $x, y \in G_1$. By minimality of G, G_1 is $L \restriction_{G_1}$ colorable. Let ϕ_1 be a coloring of G_1. Now, let w be a neighbor of v in the outer face of G_2 and consider the list assignment L' for G_2 for which $L'(v) = \{\phi_1(v)\}$, $L'(w) = c$ for some arbitrary $c \in L(w)$, and $L'(u) = L(u)\ \forall u \in V(G_2) \setminus \{v, w\}$. Note that G_2 and L' satisfy the hypothesis of the theorem. Therefore, by minimality of our counterexample, G_2 has a L'-coloring ϕ_2. But now note that, since $\phi_1(v) = \phi_2(v)$, the coloring $\phi(u) = \phi_i(u)$ if $u \in V(G_i)$ is well-defined and is an L-coloring of G, contradiction.

Hence, G is 2-connected and the outer walk C is a cycle. Now we prove that there is no chord in C. The proof is similar to the above argument. Assume there is a chord vw. Then, we have two subgraphs $G_1, G_2 \subset G$ with $G_1 \cup G_2 = G$, $G_1 \cap G_2 = \{v, w\}$ and $x, y \subset G_1$. By minimality of G, G_1 has an L-coloring ϕ_1. If we set $L'(v) = \{\phi_1(v)\}$, $L'(w) = \{\phi_1(w)\}$, and $L'(u) = L(u)$ for all other vertices $u \in V(G_2) \setminus \{v, w\}$, then G_2 is L'-colorable and a coloring of G can be constructed.

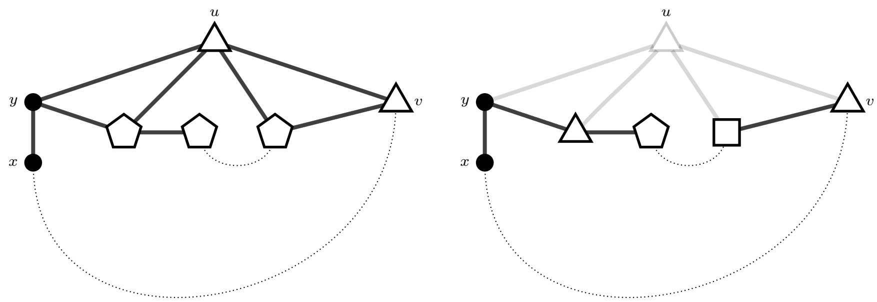


Figure 1.1: Illustration of Thomassen's reduction. At most 2 colors are erased from the lists of u's neighbors.

Now we have that G has no chords or cutvertices. Let u be the neighbor of y in the outer face other than x and let v be the neighbor of u in the outer face other than y (possibly $v = x$). Let $\{c_1, c_2\} \subseteq L(u)\ L(y)$. Now, let G' be the graph obtained by removing u from G and let L' be the list assignment for G' in which $\{c_1, c_2\}$ are removed from the lists of the neighbors of u other than v. G' satisfies the hypothesis of the theorem: every vertex in the outer face has list size at least 3, since each of those vertices is either a vertex previously in the outer face of G all of which have their previous lists (the only neighbors of u in the outer face are u and y, since G has no chords), or a previously interior vertex, which has had at most 2 of its ≥ 5 colors removed. So G' has an L'-coloring, which can be extended to an L-coloring of G by coloring u with one of c_1 or c_2 (whichever is not in use by v), contradiction.

$\square$

Criticality definition. Discuss it

In coloring problems, it is useful to consider when a precoloring of a subgraph does not *extend* to the entire graph, that is, there is no coloring of the entire graph under certain constraints which agrees with the coloring of the subgraph.

Definition 1.3.3 (Extending)**.** Let G be a graph, $T \subseteq G$ a subgraph, and L a list assignment for G. For an L-coloring ϕ of T, we say that ϕ *extends* to an L-coloring of G if there exists an L-coloring ψ of G s such that $\phi(v) = \psi(v)$ for all $v \in V(T)$.

It is also interesting to consider graphs which are critical in this setting. To do so, we use the following definition (from [8]):

Definition 1.3.4 (T-critical)**.** Let G, T, L be as above. The graph G is *T-critical with respect to* L if for every proper subgraph $G' \subset G$ such that $T \subseteq G'$, there exists an L-coloring of T that extends to an L-coloring of G', but does not extend to an L-coloring of G. If the list assignment L is clear from context, we just say *T-critical*.

Definition 1.3.5 (ϕ-critical)**.** Let, G, T, L be as above. The graph G is ϕ-critical for a coloring ϕ of T if ϕ extends to every proper subgraph of G containing T but not to G.

In a way similar to the general notion of criticality, we have that graphs for which colorings of T do not extend contain a non-trivial T-critical subgraph:

Lemma 1.3.6. *Let G be a graph, T a subgraph, and L a list assignment for G. If there is an L-coloring ϕ of T that does not extend to G, then G contains a subgraph H with $T \subsetneq H$ which is ϕ-critical, and hence also T-critical with respect to $L \restriction_H$.*

Proof. Let ϕ be the coloring of T that does not extend and let H be a minimal subgraph of G for which ϕ does not extend. Now note that H is ϕ-critical by construction. $\square$

Lemma 1.3.7. *Let G be a graph, T a subgraph, L a list assignment for G, and $H \supseteq T$ a subgraph of G which is minimal with respect to the following property: for every L-coloring ϕ of T that extends to H, ϕ also extends to G. Then H is T-critical.*

Proof. Suppose not. Then, H contains a proper subgraph H' so that every L-coloring ϕ that extends to H' also extends to H and hence to G. But then H' is a smaller subgraph with that property, contradiction. $\square$

We also have the following lemma from [8]:

Lemma 1.3.8. *Let T be a subgraph of a graph G such that G is T-critical with respect to a list assignment L. Let $A, B \subseteq G$ be such that $A \cup B = G$ and $T \subseteq A$. Then $G[V(B)]$ is $A[V(A) \cap V(B)]$-critical.*

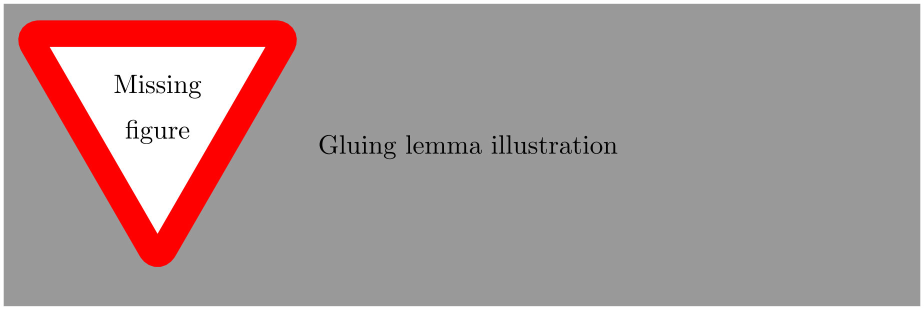
Proof. Let $G' = G[V(B)]$ and $S = A[V(A) \cap V(B)]$. If $G' = S$ there is nothing to say, suppose otherwise that $G' \neq S$ and note that therefore G' contains an edge not in S (in fact, all isolated vertices of G' must be in S). Suppose for a contradiction that G' is not S-critical. Then, by taking a maximal proper subgraph that defies the defintion, there exists an edge $e \in E(G') \setminus E(S)$ such that every L-coloring of S that extends to $G' \setminus e$ also extends to G'. Since G is T-critical and $e \notin E(T)$, there exists a coloring $\phi \restriction_T$ of T that extends to an L-coloring ϕ of $G \setminus e$, but does not extend to an L-coloring of G. However, by the choice of e, the restriction $\phi \restriction_S$ extends to an L-coloring ϕ' of G'. Let ϕ'' be such that $\phi''(v) = \phi'(v) \, \forall v \in V(G')$ and $\phi''(v) = \phi(v) \, \forall v \in V(G) \setminus V(G')$. Now, since $A \cup B = G$, ϕ'' is an L-coloring of G extending $\phi \restriction_T$, a contradiction. $\square$

For us, it will be more useful in this form:

check if this is the best phrasing

Lemma 1.3.9 (Gluing Lemma)**.** *Let T be a subgraph of a graph G such that G is T-critical with respect to a list assignment L. Let $A, B \subseteq G$ be such that $A \cup B = G$. Then $G[V(B)]$ is $(A[V(A) \cap V(B)] \cup T)$-critical.*

Proof. Apply 1.3.8 to $A' = A \cup T$ and $B' = B$. $\square$



The reason we decided to name it "Gluing lemma" in this work is that it is useful to visualize the graph G as made of two separate pieces, A and B, which are glued together along $A[V(A) \cap V(B)]$. In our approach we will frequently use the fact that all T-critical graphs can be "decomposed" in this way.

Analogous results for List coloring

Chapter 2

Critical Graphs on the Torus

Section header

2.1 An Overview of Postle's Approach

Here we briefly explain Postle's approach in [6] to obtain the result on the finiteness of 6-list-critical graphs in general surfaces, mentioning specially those intermediate results or definitions we will also use in our approach. The results obtained by Postle are very non-explicit in the sense that the (unspecified) constant in the size bounds for the graphs he obtains is extremely large and hence useless for our purpose of finding an explicit characterization. Nevertheless, given that our approach is primarily guided by this work we consider it of interest to provide a exposition of the main argument.

The results developed in [6] in 2012 have been published successively in journal articles afterwards, often with improvements in exposition or in the strength of the result. We refer to the corresponding published article in the discussion of each particular result.

2.1.1 Notation and Terminology

Postle works mainly in a setting similar to the hypothesis of Thomassen's stronger theorem: list assignments L which have list sizes of length at least 5 for interior vertices and at least 3 for exterior vertices with some exceptions. This setting is encapsulated in the concept of *canvas*.

Definition 2.1.1 (Canvas)**.** We say that (G, S, L) is a *canvas* if G is a connected plane graph with outer walk C, S is a subgraph of C, and L is a list assignment such that $|L(v)| \geq 5 \, \forall v \in V(G) \setminus V(C)$ and $|L(v)| \geq 3 \, \forall v \in V(C) \setminus V(S)$. If S is a path, we say (G, S, L) is a *path-canvas* or a *wedge*. If $S = C$ and C is a cycle, then (G, C, L) is a *cycle-canvas*.

Note: in some places like [8], the term “canvas” is used for what Postle calls in [6] “cycle-canvas”.

We can restate Thomassen’s Stronger Theorem in these terms:

Theorem 2.1.2. *If (G, P, L) is a path-canvas and $|V(P)| \leq 2$, then G is L-colorable.*

Definition 2.1.3 (Critical canvas)**.** We say that a canvas (G, S, L) is *critical* if it is S-critical with respect to L.

It is interesting to study in which circumstances can a critical canvas contain chords or cutvertices.

determine if I am going to use essential definition

Definition 2.1.4.

Lemma 2.1.5. *If $T = (G, S, L)$ is a critical canvas, then:*

1. *Every cutvertex of G is essential.*
2. *Every chord of the outer walk of G is essential.*

Proof. □

2.1.2 Variations on Thomassen’s Condition

Much of the technical work on Postle’s thesis relies in a careful study of what happens if one varies the condition on 1.3.2. One of the most elegant (and also useful) results is the following strengthening to Thomassen’s Stronger Theorem, originally conjectured by Hutchinson in [5].

Theorem 2.1.6 (Two Lists of Size Two Theorem [7])**.** *If G is a plane graph with outer cycle C, $v_1, v_2 \in C$ and L is a list assignment with $|L(v)| \geq 5$ for all $v \in V(G) \setminus V(C)$, $|L(v)| \geq 3$ for all $v \in V(C) \setminus \{v_1, v_2\}$, and $|L(v_1)| = |L(v_2)| = 2$, then G is L-colorable.*

Or, in the language of canvases:

Theorem 2.1.7. *If (G, S, L) is a canvas with $|V(S)| = 2$ and $L(v) \geq 2$ for $v \in S$, then G is L-colorable.*

This theorem is not true when one of the two vertices has list of size 1. In fact, Postle characterizes exactly when it fails:

Definition 2.1.8 (Coloring Harmonica)**.** Let G be a plane graph and L a list assignment for G. Given an edge uv and a vertex w both from the outer face of G, we say that (G, L) is a *coloring harmonica from uv to w* if either:

- G is a triangle with vertex set $\{u, v, w\}$ and $L(u) = L(v) = L(w)$ with $|L(u)| = 2$, or

- There exists a vertex z incident with the outer face of G such that uvz is a triangle in G, $L(u) = L(v) \subseteq L(z)$, $|L(u)| = |L(v)| = 2$, $|L(z)| = 3$, and the pair (G', L') is a coloring harmonica from z to w, where G' is obtained by deleting **one or both** of the vertices u, v and L' is obtained from L by $L'(z) = L(z) \setminus L(u)$ and $L'(x) = L(x)$ for all other vertices $z \neq x \in V(G')$.

Given two vertices u, w in the outer face of G, we say (G, L) is a *coloring harmonica from u to w* if there exist vertices x, y incident with the outer face of G such that uxy is a triangle in G, $|L(u)| = 1$, $L(x) - L(u) = L(y) - L(u)$, $|L(x) - L(u)| = 2$, and (G', L') is a coloring harmonica from xy to w, where G' is obtained from G by removing u, and L' is obtained from L by seting $L'(x) = L'(y) = L(x) - L(u)$ and $L'(z) = L(z)$ for every $z \in V(G') \setminus \{x, y\}$.

We say that (G, L) is a coloring harmonica if it is a coloring harmonica from uv to w or a coloring harmonica from u to w for some u, v, w as specified earlier.

See the example in (reference to figure) (from [9]) for some clarity with respect to this mutually recursive definition. Note that the definition makes it clear that graphs which contain a coloring harmonica as a subgraph are not L-colorable.

Theorem 2.1.9. *One List of Size One and One List of Size Two Theorem [9] Let G be a plane graph with outer cycle C, let $p_1, p_2 \in V(C)$, and let L be a list assignment with $|L(v)| \geq 5$ for all $v \in V(G) \setminus V(C)$, $|L(v)| \geq 3$ for all $v \in V(C) \setminus \{p_1, p_2\}$, $|L(p_1)| \geq 1$ and $|L(p_2)| \geq 2$. Then G is L-colorable if and only if the pair (G, L) does not contain a coloring harmonica from p_1 to p_2.*

Studying conditions of the sizes of the lists in the boundary in which the graph is not L-colorable like this one is also useful, because such conditions arise when dealing when reductions and therefore characterizing which are the critical graphs in such settings can give fruitful results.

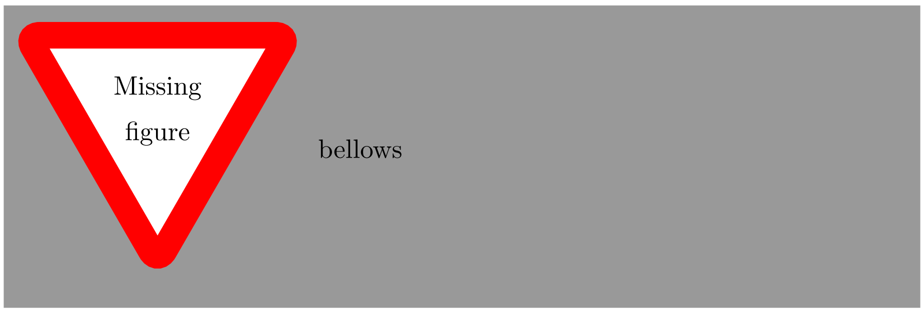
Thomassen already studied when does the coloring of a path of length 2 not extend:

Definition 2.1.10 (Bellows)**.** We say that a path-canvas (G, P, L) with $P = p_0p_1p_2$ is a *bellows* (terminology from [6]) or a *generalized wheel* (terminology from [13]) if either:

- G has no interior vertices and its edge set consists of the edges of the outer cycle plus all edges from p_1 to vertices of the outer cycle. In this case, we say that (G, P, L) is a *fan*.

- G has one interior vertex u and its edge set consists of the edges of the outer cycle plus all edges from u to vertices of the outer cycle. In this case, we say that (G, P, L) is a *turbofan*.

- G can be formed by gluing two smaller bellows from the edges p_1p_2 and p_0p_1 respectively.



Theorem 2.1.11 ([13], Theorem 3)**.** *If $T = (G, P, L)$ is a path-canvas with path length 2, then G is L-colorable unless T has a bellows as a subcanvas.*

Postle studies when the coloring of two paths of length 1 does not extend. He finds the following obstruction:

Definition 2.1.12 (Accordion)**.** We say that a canvas $T = (G, P_1 \cup P_2, L)$ with P_1, P_2 distinct paths of length 1 is an *accordion* with *ends* P_1, P_2 if T is a bellows with $P_1 \cup P_2$ path of length 2 or T is the gluing of two smaller accordions $T_1 = (G_1, P_1 \cup U, L)$ with ends P_1, U and $T_2 = (G_2, P_2 \cup U, L)$ with ends U, P_2 along a chord $U = u_1u_2$ where $|L(u_1)|, |L(u_2)| \leq 3$.

The main result he obtains is that if the two paths are sufficiently far apart, then the graph contains a proportionally large accordion or a coloring harmonica as a subgraph.

Theorem 2.1.13 (Bottleneck Theorem, loosely stated)**.** *If $T = (G, P \cup P_0, L)$ is a canvas with P, P_0 distinct edges of C with $d(P, P_0) \geq 14$, then either there exists an L-coloring of G, or there exists a subcanvas $(G_0, U_1 \cup U_2, L)$ of T where $d_{G_0}(U_1, U_2) = \Omega(d_G(P, P_0))$ which is an accordion or a coloring harmonica.*

This result, along with coloring and structural properties of accordions and harmonicas, is often used as a technical lemma when proving the following results.

2.1.3 Linear Bound on Critical Cycle-Canvases

Postle proves the following result:

Theorem 2.1.14 ([8])**.** *Let G be a plane graph with outer cycle C and L a 5-lis-assignment for G. Let H be a minimal subgraph of G such that every L-coloring of C that extends to an L-coloring of H also extends to an L-coloring of G. Then H has at most $19|V(C)|$ vertices.*

Or, equivalently stated in the language of critical canvases:

Theorem 2.1.15. *If (G, C, L) is a critical cycle-canvas, then $|V(G)| \leq 19|V(C)|$.*

The equivalence of the two statements is given by 1.3.7. This result is interesting in its own right because by 1.3.9, all faces of a T-critical graph which do not separate vertices from T are in fact critical cycle-canvases, and therefore what the result tells us is that for such graphs there is only finitely many kinds of faces that can appear for each given cycle length. This gives us a lot of information of how critical graphs look like.

A first observation that can be made is that critical cycle-canvases (in which C is indeed a simple cycle) are 2-connected, so each face is bounded by a cycle:

Lemma 2.1.16. *If (G, C, L) is a critical cycle-canvas, then it is 2-connected.*

Proof. If G is not 2-connected, then there exist subgraphs A, B such that $A \cup B = G$ with $|V(A) \cap V(B)| \leq 1$ and $|V(B) \setminus V(A)| \geq 1$. Assume $C \subseteq A$ and apply 1.3.9 to get that B is $A[V(A) \cap V(B)]$-critical, contradicting 1.3.2. □

The key result in Postle's proof of the linear bound for cycles is the following theorem about the structure of critical cycle-canvases:

Theorem 2.1.17 (Cycle Chord or Tripod Theorem)**.** *If (G, C, L) is a critical cycle-canvas, then either*

1. *C has a chord in G, or*
2. *there exists a vertex $v \in V(G)\ V(C)$ with at least three neighbors on C such that at most one of the faces of $G[\{v\} \cup V(C)]$ includes a vertex or edge of G.*

think if I should include proof of CCTT

Using this result, Postle carefully examines what happens near the boundary cycle in order to define some quantities related to sums of lengths of faces and proves that certain inequalities with those quantites are mantained when adding tripods in critical canvases.

2.1.4 The Two Precolored Triangles Theorem

Next, Postle proves the following theorem:

Theorem 2.1.18. *There exists d such that the following holds. Let G be a planar graph and T_1, T_2 triangles in G at distance at least d. Let L be a 5-list-assignment of G. Then, every L-coloring of $T_1 \cup T_2$ extends to an L-coloring of G.*

The value of d that Postle obtains is not explicitly stated, but it is on the order of 100. However, we conjecture that 4 or 5 suffices.

Conjecture 2.1.19. *Let G be a planar graph and T_1, T_2 triangles in G at distance at least 5. Let L be a 5-list-assignment of G. Then, every L-coloring of $T_1 \cup T_2$ extends to an L-coloring of G.*

The argument that Postle uses to prove his result is as follows. First, he proves that one can precolor a path between the two triangles in such a way that, when deleting the path and deleting the corresponding colors from the lists of neighboring vertices, all remaining non-precolored vertices have lists of size at least 3. The proof of this begins with the simple observation that each vertex outside a shortest path has at most 3 neighbors inside the path. Using planarity properties, a shortest path can be found so that it can be colored in such a way that the vertices with 3 neighbors inside the path only see two different colors from their lists.

After precoloring and deleting the path between the two triangles, a canvas $(G, P_1 \cup P_2, L)$ is obtained. If there was a precoloring of the triangles that did not extend, then the canvas contains a critical canvas, and by 2.1.13 it contains a proportionally long accordion or harmonica. Postle proves that this (together with some technical details related to how the path between the triangles was chosen) implies that in the original graph there must be a long chain of separating triangles so that the graph between each separating triangle pertains to one of three very specific types, which he calls tetrahedral, octahedral or hexadecahedral bands.

check if I defined separating triangle in introduction

Finally, he proves that for a sufficiently long chain of this type, any precoloring of the innermost and outermost triangles extends to the whole chain. This proves the theorem, because of the following observation:

Proposition 2.1.20. *Let G be a plane graph with L a list assignment, T_1, T_2 two facial triangles with T_1 bounding the infinite face of G, and T_1', T_2' two triangles such that T_1' is a separating triangle between T_1 and T_2' and T_2' is a separating triangle between T_1' and T_2. Denote by $G[T_1', T_2']$ the subgraph comprised between the two triangles T_1', T_2'. If there exists some L-coloring of $T_1 \cup T_2$ that does not extend to G, then there exists some L-coloring of $T_1' \cup T_2'$ that does not extend to $G[T_1', T_2']$.*

Proof. By 1.3.2, the coloring on T_1 extends to $G[T_1, T_1']$ and the coloring on T_2 extends to $G[T_2', T_2]$. The coloring of $T_1' \cup T_2'$ given by this extensions can not extend to $G[T_1', T_2']$ by the assumption that the original coloring of $T_1 \cup T_2$ did not extend to G. □

2.1.5 Hyperbolicity

hyperbolicity and cylinder canvases section

2.2 Critical Graphs on the Torus for (usual) Vertex Coloring

In this section we discuss the result from Thomassen in [11] that characterizes the critical graphs for 5-coloring (not 5-list-coloring) on the torus.

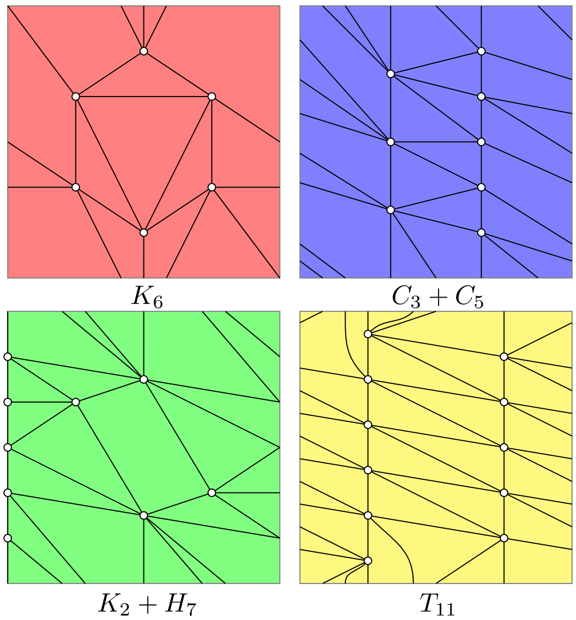


Figure 2.1: 6-critical graphs embedded on the torus.

2.2.1 The Critical Graphs

Theorem 2.2.1 ([11])**.** *A graph G embeddable on the torus is 5-colorable if and only if it does not contain the following subgraphs:*

- K_6.
- $C_3 + C_5$.
- $K_2 + H_7$, *where H_7 is the* Moser spindle, *the graph obtained by applying the Hajós construction to a pair of K_4.*

check if I explained Hajós construction in original

- T_{11}, *where T_{11} is a triangulation of the torus with 11 vertices.*

Where + denotes the join of two graphs: their disjoint union with all pairs of vertices from different graphs joined by edges.

If a graph is not 5-colorable, it is not 5-list-colorable, so all graphs that contain any of the above subgraphs are not 5-list-colorable. We conjecture that this characterizes the 5-list-colorable graphs on the torus too:

Conjecture 2.2.2. *A graph G embeddable on the torus is 5-list-colorable if and only if it does not contain the following subgraphs: K_6, $C_3 + C_5$, $K_2 + H_7$, T_{11}.*

This means that those are the minimal 6-list-critical graphs on the torus. Note that there may be additional 6-list-critical graphs embeddable on the torus, but what we are conjecturing is that they all contain those subgraphs. For example:

Observation 2.2.3. *K_7 is 6-list-critical.*

Proof. Consider the following 5-list-assignment for K_7: $L(v_1) = L(v_2) = L(v_3) = L(v_4) = L(v_5) = \{1,2,3,4,5\}$, $L(v_6) = L(v_7) = \{1,2,3,4,6\}$. K_7 is not L-colorable, since there are only 6 available colors. But any subgraph is L-colorable. Let's give a coloring ϕ for $K_7 \setminus v_iv_j$. If $i, j \leq 5$, then setting $\phi(v_i) = \phi(v_j) = 5$ and $\phi(v_7) = 6$ leaves 4 vertices to be colored with 4 colors. If $i \leq 5$ and $j \geq 6$, then setting $\phi(v_i) = \phi(v_j) = 1$, $\phi(v_{13-j}) = 6$ leaves 4 vertices to be colored with 4 colors. If $\{i, j\} = \{6, 7\}$, then $\phi(v_i) = \phi(v_j) = 6$ leaves 5 vertices to be colored with 5 colors.

Hence, K_7 is L-critical for a 5-list-assignment L, and is therefore 6-list-critical. □

2.2.2 An Overview of Thomassen's Approach

Thomassen's article where he characterizes the graphs on the torus ([11]) predates his result on finitely many 6-critical graphs for all surfaces ([12]). For the characterization of 6-critical graphs on the torus, he only uses elementary, relatively straighforward arguments that work on specifically in the torus. We briefly summarize his approach here in order to discuss which arguments can be reused for the list-coloring case.

First, Thomassen considers the case when the minimum degree is at least 6. He obtains this result:

Theorem 2.2.4. *Let G be a graph embedded on the torus with $\delta(G) \geq 6$. Then G is 5-colorable unless $G = K_7$ or $G = T_{11}$.*

We will discuss later how to arrive at this result, because this is the part of the proof that can be adapted for list-coloring. But let us first describe Thomassen's argument for general graphs.

He assumes a minimum counterexample G_0 to 2.2.1 (the counterexample has minimum number of vertices, maximum number of edges restricted to that, and some other assumptions about details we will not discuss here). By the previous result, there must be a vertex $v_0 \in V(G_0)$ with degree ≤ 5, and the degree of v_0 is in fact equal to 5 by minimality of the counterexample.

Consider two vertices $x, y \in N(v_0)$ which are not adjacent (if all the vertices of $N(v_0)$ were adjacent, then G_0 would contain K_6, a contradiction). Let G_{xy} be the graph obtained from $G_0 \setminus v_0$ by identifying the vertices x and y. G_{xy} can be embedded in the torus by modifying the embedding of G_0. If G_{xy} were 5-colorable, then we would have a 5-coloring of G_0 by assigning the same color to x and y and coloring v_0 with a color not appearing in its 5 neighbors. Hence, G_{xy} is not 5-colorable and by minimality of our counterexample it contains K_6, $C_3 + C_5$, $K_2 + H_7$ or T_{11}.

The above argument works for all pairs x, y of non-adjacent vertices in $N(v_0)$, so potentially we can have many different obstructions for each of the corresponding G_{xy} subgraphs. But we can prove that, by minimality, there can not be much else in G_0 apart from these obstructions arising from all the G_{xy} subgraphs. More precisely:

Proposition 2.2.5. *For any non-adjacent $x, y \in N(v_0)$, let G'_{xy} a copy of K_6, $C_3 + C_5$, $K_2 + H_7$ or T_{11} in G_{xy}, and let G''_{xy} be the induced subgraph of G_{xy} by the vertex set of G'_{xy}. Then G_0 consists of v_0, $N(v_0)$, the edges between vertices of $\{v_0\} \cup N(v_0)$, and the union over all non-adjacent $x, y \in N(v_0)$ of the graph obtained from G''_{xy} by splitting the contracted vertex into x and y.*

Proof. We will prove that the graph described above, which is a subgraph of G_0, is not 5-colorable. This means, by the assumptions of minimality of vertices and maximality of edges, that G_0 is in fact equal to that subgraph.

If the subgraph had a 5-coloring, then two non-adjacent vertices x, y of $N(v_0)$ would have the same color. But by then identifying the two vertices we can get a 5-coloring of G''_{xy}, which contains the non-5-colorable subgraph G'_{xy}, contradiction.

□

Note that, since the maximum number of vertices in a critical graph is 11, this means that G_0 has at most $(11 - 1) \cdot \binom{5}{2} + 6 = 106$ vertices, and hence what remains is a finite problem.

Thomassen uses some more arguments to narrow down the remaining possibilities for G_0, but we can already see an important point of failure of this argument for list-coloring: in the proof of 2.2.5, it is used that a necessary and sufficient condition for a coloring of $G_0 \setminus v_0$ to extend to v_0 is that two neighbors of v_0 have the same color. In list coloring, this condition is not necessary. So we cannot conclude that the minimum counterexample is the union of the graphs induced by the obstructions in G_{xy} and the argument breaks down here.

2.2.3 6-Regular 6-Critical Graphs

As we said before, Thomassen's argument for graphs with minimum degree 6 can be reused for list-coloring. This is because these graphs are very restricted, and therefore their structure can be completely characterized and a 5-(list)-coloring can be explicitly exhibited for the ones that are colorable. The basic result is this.

Proposition 2.2.6. *If a graph G embedded on the torus has $\delta(G) \geq 6$, then:*

1. *G is 6-regular.*
2. *G is a triangulation of the torus.*

Proof. We apply Euler's formula: let V, E, F be the number of vertices, edges and faces in the embedding, respectively. We have that $\delta(G) \geq 6 \implies V \leq \frac{1}{3}E$

with equality iff G is 6-regular, and $F \leq \frac{2}{3}E$ with equality iff G is a triangulation. Then $0 = V - E + F \leq \frac{1}{3}E - E + \frac{2}{3}E = 0$, so we have equality on both inequalities. $\square$

Using the proposition above, Thomassen then proves the following:

Proposition 2.2.7 (3.2 in [11]). *Let G be a 6-regular graph on the torus. If G contains a vertex v, such that $\{v\} \cup N(v)$ induces a nonplanar graph, then $G = K_7$ or G is obtained from K_8 or K_9 by deleting the edges of a 1-regular or 2-regular subgraph.*

The study of 6-regular graphs on the torus without vertices whose neighborhood induces a nonplanar graph was already done by Thomassen in his previous paper [14], in the context of finding all tilings of the torus in order to prove a conjecture by Babai about vertex-transitive graphs.

finish 6-regular graphs section

see if it is worth it to include, maybe it is nontrivial to actually prove 5-list-colorability

2.3 Our Approach

Chapter 3

Generation of Critical Graphs

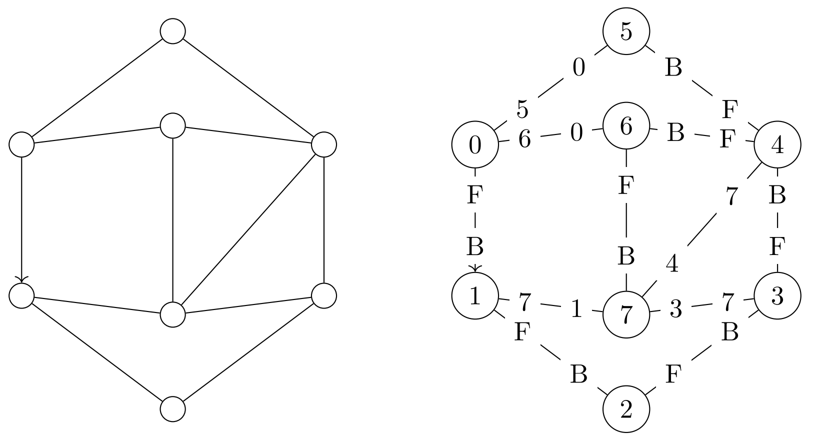
In this chapter we describe algorithms for processing and generating critical canvases via computer search.

3.1 Representation of Canvases

The first step is deciding how to represent canvases in our algorithms. Recall that a canvas T is a tuple (G, S, L) where G is a plane graph, S is a subgraph of the outer face and L is a list assignment for G satisfying some conditions. Here, in each algorithm we will usually be working with one particular family of canvases at a time, for example cycle-canvases or path-canvases with a fixed size of cycle or path, so the information about the subgraph S can be "implicit" in each different representation for each different algorithm instead of working with a general representation that allows all canvases. Also, in some scenarios we will be working with conditions on the list assignment L which are different from the ones in the definition of canvas. In this section we intend to just expose some general ideas about how the representation of graphs in this context can be done, which will be afterwards applied in different scenarios.

The most important thing to state is that we will not be interested in storing the list assignment L at all. This is because there is a significant combinatorial explosion in the number of list assignments to be considered and we are interested in the graphs themselves, not the list assignments. Also, most of the results we will be using such as 1.3.9 or 2.1.17 are directly related to subgraphs and not list assignments, and while they are in theory stated with respect to a fixed list assignment, it is more useful in practice to not consider the list assignment at all.

Thus, when we generate all critical canvases (G, S, L), what we will actually be doing is generate all pairs (G, S) such that there exists some list assignment L so that (G, S, L) is a critical canvas. In some scenarios, we will also be interested



$\Longrightarrow$ FFFFF0BF0F134BB7B7BB7B65

Figure 3.1: Example of a DFS traversal transcript for a graph. It is the lexicographically smallest transcript in this case.

in storing the prescribed size of the list assignment for each vertex: that is, we will be storing a tuple (G, S, f) with $f : V(G) \to \mathbb{N}$ so that we will only be considering list assignments L with $|L(v)| = f(v)$, but other than that we will not store information about the actual list assignment.

We store the information of the graph G with an adjacency list. We will also be interested in storing the planar embedding of the graph: to do so, we order the edges in the adjacency list of each vertex according to their clockwise order in the embedding (as in a *rotation system*). This information, together with the information of which vertices are in the outer face, is enough to reconstruct the embedding.

We will want to test when two canvases are isomorphic. More generally, we will want to have a canonical form for each canvas, so that given a set of canvases $\mathcal{S}$ and a new canvas T, we can check whether there is a canvas isomorphic to T in $\mathcal{S}$ by checking the presence of the corresponding canonical form of T in an associative array with the canonical forms of the canvases in $\mathcal{S}$.

In order to produce the canonical form, define the *transcript of the DFS traversal starting at edge* $u \to v$ as the string generated by procedure (REF ALGORITHM). That is, we do a depth-first traversal of the graph following the edges on each vertex in clockwise order, assigning labels to vertices based in the order in which we first visit them and storing information for each edge we visit in the traversal: `F` for edges towards a new vertex in the traversal, `B` for edges towards the immediately previous vertex in the traversal stack (which signifies the end of the edges for the current vertex and the return to the previous vertex of the stack) and the label of the other endpoint for other edges. See (REF ALGORITHM) for details.

Write canvas canonization algorithm

We compute such string for all the edges of the outer face as starting edges, and we take the lexicographically smallest one as the canonical form. It is clear that two plane graphs have the same canonical string if and only if they have isomorphic (in terms of the rotation system) embeddings.

Maybe be more explicit with the explaination of the canonical form, and indeed prove that it is canonical

.

3.2 Generation of Critical Cycle-Canvases

Our algorithm for the generation of critical cycle-canvases is based on 2.1.17. This theorem says that every critical cycle-canvas can either be decomposed into two smaller critical cycle-canvases through a chord in the outer face, it can be decomposed into a “tripod”, a vertex v with at least 3 neighbors in C, and a smaller critical cycle-canvas contained in the only nonempty face incident with v. In these decompositions, it is possible that instead of a smaller critical canvas we get an empty canvas, which is technically not critical.

check that all definitions of critical canvases are consistent with this

This implies that we can generate all critical cycle-canvases from smaller cycle-canvases by gluing cycle-canvases through outer face edges to get a canvas with a chord, or by adding a tripod to the outside of a cycle-canvas. We then have to check whether the resulting canvas is indeed critical, since the decomposition into two smaller critical cycle-canvases is a necessary but not sufficient condition for criticality. We will see how to do this in Section 4. .

references for sections

The three cases for generation of critical cycle-canvases

figure for generation of critical cycle-canvases and references to figure

If we are generating cycle-canvases with cycle length ℓ, then a chord partitions the cycle-canvas into two cycle-canvases of length a, b with $a, b \geq 3$ and $a + b = \ell + 2$ (see figure (a)), so $a, b \leq \ell - 1$ and therefore if we have generated all cycle-canvases with cycle length $< \ell$ we can generate all cycle-canvases with cycle length ℓ with a chord. In the case of adding a tripod, though, if the vertex v of the tripod is adjacent to only three adjacent vertices in the outer face, then the smaller cycle-canvas has the same cycle length as the larger cycle-canvas (see figure (c)).

In order to resolve this, what we do is first generate all the cycle-canvases obtained from cycle-canvases with smaller cycle size (as in figure (a) and (b)), enqueue the resulting critical canvases, and then process the canvases from the queue and add tripods to three consecutive vertices in all possible ways, enqueueing the new critical cycle-canvases that are found. Here is the description of the algorithm:

Fix lines with comments having semicolon in algorithms

Algorithm 1: Generation of Critical Cycle-Canvases.

```
/* Generate critical canvases of cycle size ℓ, including
empty one                                                  */
function generateCriticalCycleCanvases(ℓ)
    for i = 3, ..., ℓ − 1 do
        S_i ← generateCriticalCycleCanvases(i);
    end
    S ← {emptyCycle(ℓ)};
    for a = 3, ..., ℓ − 1 do
        b ← ℓ − a + 2;
        for G_1 ∈ S_a do
            for G_2 ∈ S_b do
                T ← fuseChordSet(G_1, G_2);
                ; /* Set of cycle-canvases obtained by fusing G_1
                 and G_2 along outer cycle edges in all possible
                 ways */
                for G ∈ T do
                    if G ∉ S AND isCritical(G) then
                        S ← S ∪ {G};
                    end
                end
            end
        end
    end
    for k = 3, ..., ℓ − 1 do
        for G_1 ∈ S_k do
            T ← addTripodSet(G_1, ℓ − k + 3, 3);
            ;     /* Set of cycle-canvases obtained by adding a
             tripod with 3 neighbors in the outer face to get a
             cycle-canvas of length ℓ in all possible ways */
            for G ∈ T do
                if G ∉ S AND isCritical(G) then
                    S ← S ∪ {G};
                end
            end
        end
    end
    Q ← Queue(S);
    while Q is not empty do
        G_1 ← first(Q);
        dequeue(Q);
        T ← addTripodSet(G_1, 3, 3);
        for G ∈ T do
            if G ∉ S AND isCritical(G) then
                S ← S ∪ {G};
                enqueue(Q, G);
            end
        end
    end
    return S;
end
```


fix algorithm size

Note that we only need to add tripods with 3 adjacent neighbors since vertices with a larger number of neighbors in the outer face can be obtained by first adding chords and then adding finally adding a tripod with 3 neighbors. However, often we are interested in just generating chordless critical canvases. In that case, we do need to add tripods of all sizes. The modified algorithm for chordless critical cycle-canvases is the following:

Algorithm 2: Generation of Chordless Critical Cycle-Canvases.

```
/* Generate chordless critical canvases of cycle size ℓ,
including empty one                                        */
function generateChordlessCriticalCycleCanvases(ℓ)
    for i = 3,...,ℓ − 1 do
        S_i ← generateChordlessCriticalCycleCanvases(i);
    end
    S ← {emptyCycle(ℓ)};
    for k = 3,...,ℓ − 1 do
        for j = 3,...,ℓ − k + 3 do
            for G_1 ∈ S_k do
                T ← addTripodSet(G_1, ℓ − k + 3, j);
                for G ∈ T do
                    if G ∉ S AND isCritical(G) then
                        S ← S ∪ {G};
                    end
                end
            end
        end
    end
    Q ← Queue(S);
    while Q is not empty do
        G_1 ← first(Q);
        dequeue(Q);
        T ← addTripodSet(G_1, 3, 3);
        for G ∈ T do
            if G ∉ S AND isCritical(G) then
                S ← S ∪ {G};
                enqueue(Q, G);
            end
        end
    end
    return S;
end
```

3.3 Generation of Critical Wedges

We are will be not only interested in generating critical cycle-canvases, but also critical path-canvases or wedges. There are infinitely many of those for path length greater than 1, but as we will see in coming sections, we will be able to have a finite number of them if we impose additional conditions.

Fortunately, we also have an analogue of theorem 2.1.17 for wedges:

Theorem 3.3.1 (Wedge Chord or Tripod Theorem)**.** *If (G, P, L) is a 2-connected critical wedge, then either*

1. *The outer walk C has a chord in G, or*

2. *there exists a vertex $v \in V(G)\ V(P)$ with at least three neighbors on P such that at most one of the faces of $G[\{v\} \cup V(P)]$ includes a vertex or edge of G.*

Proof. Assume not. Then, we will show that every L-coloring of P extends to an L-coloring of G, contradiction. Let ϕ be any L-coloring of P, $G' = G \setminus P$ and $L'(v) = L(v) \setminus \{\phi(u) : u \in V(P) \text{ neighbor of } v\}$ for each $v \in V(G')$. Note that $|L'(v)| \geq 5$ for every interior vertex of G'. Let C' be the outer walk of G' and let v_1, v_2 be the two vertices of G' that were adjacent to the two endpoints of P in G. Note that $|L'(v)| \geq 3$ for all $v \in C' \setminus \{v_1, v_2\}$ since G was 2-connected and had no chords, and $|L(v_1)|, |L(v_2)| \geq 2$. Hence, G' is L'-colorable by 2.1.6. □

(This version of the theorem is slightly different than the one proved by Postle in [6]).

Based on this theorem, we can design an algorithm to generate critical wedges similar to the one used to generate critical canvases. The main additional consideration we need to take into account is that now we need to generate non-2-connected critical wedges by gluing smaller critical wedges along the cutvertices. Observe that the cutvertices must necessarily be part of P. Fix critical wedges algorithm

Algorithm 3: Generation of Critical Wedges.

```
/* Generate critical wedges of path size ℓ, including the
path                                                              */
function generateCriticalWedges(ℓ)
    for i = 3, ..., ℓ − 1 do
        S_i ← generateCriticalCycleCanvases(i);
    end
    S ← {emptyPath(ℓ)};
    for a = 3, ..., ℓ − 1 do
        b ← ℓ − a + 2;
        for G_1 ∈ S_a do
            for G_2 ∈ S_b do
                T ← fuseChordSet(G_1, G_2);
                ; /* Set of cycle-canvases obtained by fusing G_1
                 and G_2 along outer cycle edges in all possible
                 ways */
                for G ∈ T do
                    if G ∉ S AND isCritical(G) then
                        S ← S ∪ {G};
                    end
                end
            end
        end
    end
    for k = 3, ..., ℓ − 1 do
        for G_1 ∈ S_k do
            T ← addTripodSet(G_1, ℓ − k + 3, 3);
            ;      /* Set of cycle-canvases obtained by adding a
             tripod with 3 neighbors in the outer face to get a
             cycle-canvas of length ℓ in all possible ways */
            for G ∈ T do
                if G ∉ S AND isCritical(G) then
                    S ← S ∪ {G};
                end
            end
        end
    end
    Q ← Queue(S);
    while Q is not empty do
        G_1 ← first(Q);
        dequeue(Q);
        T ← addTripodSet(G_1, 3, 3);
        for G ∈ T do
            if G ∉ S AND isCritical(G) then
                S ← S ∪ {G};
                enqueue(Q, G);
            end
        end
    end
    return S;
end
```

Chapter 4

Criticality Testing

In this chapter we describe algorithms used to determine list-criticality of graphs. Recall that we are not storing the explicit list assignment L for our graphs, so what we want to check is whether there exists a L so that the graph is critical with respect to that L. However, even if L was fixed, it would still be a computationally hard problem to determine criticality. What we will do instead is check for weaker properties, and therefore admit some false positives, that is, some graphs we identify as list-critical for which actually no suitable L exist. Our hope is that the tests will be exhaustive enough so that finiteness results such as 2.1.14 still hold for the weaker properties we are testing, and our algorithms terminate. We will see that indeed, the algorithms described here work very well in practice at discarding non-critical graphs.

explain that we work with graphs with prescribed list sizes

4.1 Degree Properties

We can start with an easy observation:

Observation 4.1.1. *In a f-list-critical graph, $d(v) \geq f(v)$ for all vertices v.*

So if we find a vertex with degree less than the prescribed list size, we can conclude that the graph is not list-critical. However, this is a very weak test. We can incorporate another test concerning the vertices with $d(v) = |L(v)|$: there is the following result by Gallai showing that the subgraph induced by those vertices must have a certain structure, generalizing the classical Brooks theorem for vertex coloring:

Theorem 4.1.2 (Gallai [3])**.** *Let G be a f-list-critical graph and let H be the subgraph of H induced by the vertices with $d(v) = f(v)$. Then each 2-connected component of H is a complete graph or an odd cycle.*

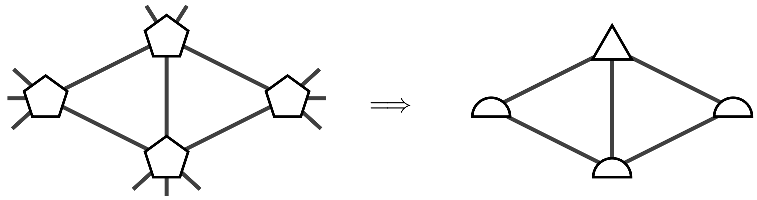


Figure 4.1: Illustration of **??**: from a subgraph, we get a graph with prescribed list sizes.

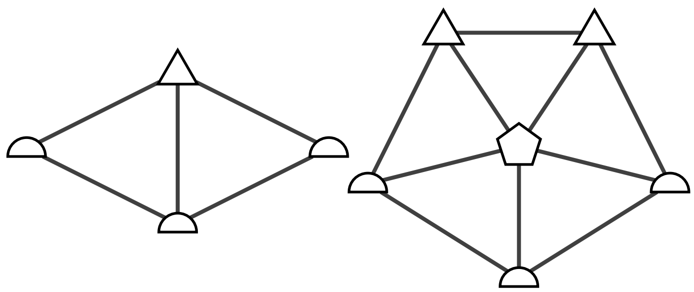


Figure 4.2: Some colorable reducible configurations.

4.2 Reducible Configurations

The *method of reducible configurations* is a usual technique in graph coloring problems. It consists in identifying subgraphs or other structures that can not appear in critical graphs. We have the following observation:

Proposition 4.2.1. *Let G be an f-list-critical graph, and let H be an induced subgraph of G such that $\forall v \in H, f(v) > d_{G \setminus H}(v)$, where $d_{G \setminus H}(v)$ is the number of neighbors of v which are not in H. Then H is not g-list-colorable, where $g(v) = f(v) - d_{G \setminus H}(v)$.*

Proof. If $G = H$, it is immediate. Assume $H \subsetneq G$, and let G be L-critical. Let ϕ be a coloring of $G \setminus H$, and let L' be the g-list-assignment of H given by $L'(v) = L(v) \setminus \{\phi(u) : u \in N_G(v), u \notin H\}$. Then H is not L'-colorable: since otherwise, the L-coloring ϕ of $G \setminus H$ would extend to G, contradiction. $\square$

We can consider 4.1.1 to be a particular case of 4.2.1. In **??** we see a couple of examples of small graphs that are always f-colorable, so one possible test we can add to our criticality testing procedure is to search for occurrences of those graphs as induced subgraphs, and if one is found then conclude that the graph is not critical.

However, there are also reducible configurations which are not f-colorable.

Definition 4.2.2. A graph G is said to be *f-reducible* if there is a proper

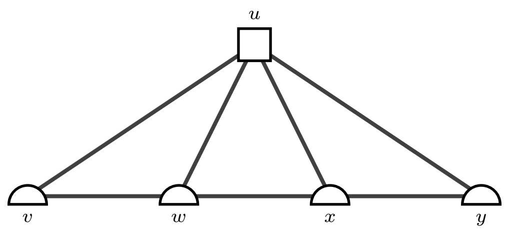


Figure 4.3: A non-colorable reducible configuration.

subgraph $H \subsetneq G$ so that for all f-list-assignments L of G, if H is $L\restriction_H$-colorable then G is L-colorable.

Proposition 4.2.3. *Let G be an f-list-critical graph, and let H be an induced subgraph of G such that $\forall v \in H, f(v) > d_{G\setminus H}(v)$, where $d_{G\setminus H}(v)$ is the number of neighbors of v which are not in H. Then H is not g-reducible, where $g(v) = f(v) - d_{G\setminus H}(v)$.*

Proof. Assume not, and let H' be the corresponding subgraph of H for which all g-colorings extend. Let G be L-critical. There exists an L-coloring ϕ of $G\setminus(H\setminus H')$. Let L' be the g-list-assignment of H given by $L'(v) = L(v)\setminus\{\phi(u) : u \in N_G(v), u \notin H\}$. Note that $\phi\restriction_{H'}$ is an L'-coloring of H', so there must be an L'-coloring ψ of H. But then the coloring Φ given by $\Phi(v) = \psi(v)$ for $v \in H$, $\Phi(v) = \phi(v)$ for $v \in G \setminus H$ is a L-coloring of G, contradiction. □

Observation 4.2.4. *The graph depicted in 4.3 with prescribed list sizes f is f-reducible.*

Proof. Let us characterize the f-list-assignments L for which the graph is not L-colorable. First, note that if $L(v) \neq L(w)$ or $L(x) \neq L(y)$ one can precolor both w and x so that the resulting graph with the corresponding colors removed from the lists is a path with list sizes at least 2 in all vertices except in one endpoint, with list size at least 1, so it is colorable. Therefore, we have $L(v) = L(w)$ and $L(x) = L(y)$. Then, note that $L(v), L(x) \subseteq L(u)$, since otherwise by coloring one vertex with a color not in $L(u)$ one can always color the graph. Therefore, the f-list-assignments for which the graph is not L-colorable are those of the form $L(u) = \{A, B, C, D\}$, $L(v) = L(w) = \{A, B\}$, $L(x) = L(y) = \{C, D\}$. But for those list assignments the graph without edge wx is also not L-colorable, so for any f-list-assignment L the graph is L-colorable if and only if the subgraph without the edge wx is L-colorable. □

4.3 The Alon-Tarsi Method

While checking for the small reducible configurations we found in the previous section is helpful, it is not good enough, because there are larger graphs which are always f-colorable but do not contain any of the reducible configurations. One could augment the list of configurations to check by manually those graphs when they are encountered, but this is ineffective and also inefficient because induced subgraph isomorphism testing starts being very expensive with larger subgraphs. We would, then, like to have a systematic method to find when graphs with prescribed list sizes f are f-list-colorable. Alon and Tarsi provided a useful criterion:

Theorem 4.3.1 (Alon-Tarsi, [2])**.** *Let G be a directed graph on vertices $v_1, \ldots, v_n$, and let L be an assignment of lists to vertices of G such that $|L(v_i)| \geq d^+(v_i)+1$ for $i = 1, \ldots, n$. If G has a different number of even and odd spanning eulerian subgraphs, then G has an L-coloring.*

Define directed graphs and d+ either here or in the introduction

.

Here *even* and *odd* eulerian subgraphs refer to the number of edges. We will explain the proof of this theorem here, since it will give us insight into how to implement it in a more efficient way than enumerating all eulerian subgraphs. For the proof we will need an algebraic result known as Combinatorial Nullstellensatz.

Theorem 4.3.2 (Combinatorial Nullstellensatz [1])**.** *Let $\mathbb{K}$ be a field and $p(x_1, \ldots, x_n) \in \mathbb{K}[x_1, \ldots, x_n]$ be a nonzero polynomial and let $t_1, \ldots, t_n$ be nonnegative integers such that the degree of p is $t_1 + \ldots + t_n$ and the coefficient of $\prod_{i=1}^{n} x_i^{t_i}$ in p is nonzero. Let $S_1, \ldots, S_n$ be subsets of $\mathbb{K}$ such that $|S_i| \geq t_i + 1$. Then there exist $a_1 \in S_1, \ldots, a_n \in S_n$ such that $p(a_1, \ldots, a_n) \neq 0$.*

4.3.1 Proof of the Alon-Tarsi Theorem

Definition 4.3.3. For a directed graph G with n vertices $v_1, \ldots, v_n$, we define its *graph polynomial* $p_G(x_1, \ldots, x_n)$ as:

$$p_G(x_1, \ldots, x_n) = \prod_{\overrightarrow{v_i v_j} \in E(G)} (x_j - x_i).$$

Observation 4.3.4. *If we associate each color with a different number (we work in, say, $\mathbb{C}$), then ϕ is a proper coloring of G if and only if $p_G(\phi(v_1), \ldots, \phi(v_n)) \neq 0$.*

Proposition 4.3.5. **??** *Let G be as in the hypothesis of the Alon-Tarsi theorem. If the coefficient of p_G at $\prod_{i=1}^{n} x_i^{d^+(v_i)}$ is non-zero, then G has an L-coloring.*

Proof. The total degree of p_G is $|E(G)| = d^+(v_1) + \ldots + d^+(v_n)$. By the Combinatorial Nullstellensatz, one can find $\phi(v_1) \in L(v_1), \ldots, \phi(v_n) \in L(v_n)$ so that $p_G(\phi(v_1), \ldots, \phi(v_n)) \neq 0$. $\square$

Observation 4.3.6. *Let G be a directed graph, and denote by $x_G = \prod_{\overrightarrow{v_i v_j} \in E(G)} x_j$. Let D_1 and D_2 be two orientations of a graph, denote by $|D_1 \Delta D_2|$ the number of edges with different direction in the two orientations. We have:*

$$p_G(x_1, \ldots, x_n) = \sum_{D \text{ orientation of } G} (-1)^{|D \Delta G|} x_D \tag{4.1}$$

Proposition 4.3.7. *The absolute value of the coefficient of* p_G *at* $\prod_{i=1}^{n} x_i^{d^+(v_i)}$ *is the difference between odd and even eulerian spanning subgraphs of* G.

Proof. Consider the set $\mathcal{S} = \{D \text{ orientation of } G : x_D = \prod_{i=1}^{n} x_i^{d^+(v_i)}\}$, that is, the set of orientations which have the same indegrees as G. We claim that this set is in bijection with the set of eulerian spanning subgraphs of G, with a bijection maps orientations with even $|D\Delta G|$ to even spanning subgraphs and orientations with odd $|D\Delta G|$ to odd spanning subgraphs. This implies the result by (4.1).

The bijection is as follows: for each $D \in \mathcal{S}$ map it to the subgraph with edges given by the edges in G which have opposite orientation as edges in D. Since the indegrees of D and G are the same, this subgraph is eulerian, and this map is clearly invertible and hence a bijection. □

This concludes the proof of 4.3.1.

4.3.2 Implementation of the Alon-Tarsi Method

Here we will explain how to use 4.3.1 in practice to check the f-colorability of a graph. The first thing we have to note is that 4.3.1 is stated with respect to a directed graph, but this is immaterial to our needs. We only have the prescribed list sizes f, and for each such prescription there can be different orientations of that satisfy the $f(v) \geq d^+(v) + 1$ condition, and we could apply the theorem to each of those. So instead of using the combinatorial characterization of 4.3.1, what we will do is directly compute the polynomial p_G with respect to an arbitrary orientation of the graph, and if any monomial $\prod x_i^{e_i}$ with $e_i < f(i)$ has a nonzero coefficient we will conclude f-colorability.

It will be convenient to think of the computation of the polynomial p_G using the expression (4.1). This way, we can compute the coefficients by enumerating all orientations of the graph and summing the signs. The implementation is as follows:

Algorithm 4: Naive Alon-Tarsi.

```
; /* Returns true if the Alon-Tarsi method determines that G
 is f-colorable. */
function alonTarsi(G, f)
    S ← allOrientations(G);
    ; /* allOrientations(G) generates all orientations of G
     (e. g. by orienting each edge by recursive
     backtracking) */
    p_G ← emptyAssociativeArray();
    ; /* We represent the polynomial p_G by an associative
     array mapping the array of n integers e_1, ..., e_n to the
     coefficient of ∏ x_i^{e_i} initialized to 0 on all values. */
    for D ∈ S do
        s ← 1;
        e ← array(n);
        for {u, v} ∈ E(G) do
            ; /* We pick an arbitrary order for u, v in each edge
             in order to determine an arbitrary orientation of
             G (e. g. set u < v in the integer labeling we
             use to represent G). */
            if uv→ ∈ E(D) then
                e_v ← e_v + 1;
            else
                s ← −s;
                e_u ← e_u + 1;
        end
        p_G[e] ← p_G[e] + s;
    end
    for (e, c_e) ∈ p_G do
        ; /* Iterate over the coefficients of the polynomial
         and if there is some nonzero coefficient of an
         appropiate monomial, return success. */
        if e_v < f(v) ∀v ∈ V(G) then
            if c_e ≠ 0 then
                return true;
            end
        end
    end
    return false;
end
```

However, there are multiple improvements that can be made over this naive implementation.

The most immediate one is that we only care about the coefficients of the monomials corresponding to orientations with indegrees less than $f(v)$ in each vertex f (we call such orientations f-orientations). Therefore we can store only coefficients of the polynomial corresponding to f-orientations. We would also like to generate only f-orientations, so that we do not have to iterate over all $2^|E(G)|$ orientations of the graphs.

If we generate orientations by a orienting each edge one by one in a recursive backtracking fashion, we want to know when to cut a branch that is not going to lead to an f-orientation. The easiest way is to cut a branch when the branch trivially does not correspond anymore to an f-orientation, that is, when the indegree of some of the vertices already reaches $f(v)$. It can be done in a more sophisticated way by reducing the problem of checking whether a partial orientation can be extended to an f-orientation to a problem of maximum bipartite matching. This way, all the branches that do not lead to an f-orientation can be immediately discarded by this test, and we can obtain an enumeration of all f-orientations in polynomial time for each f-orientation.

However, it turns out it is more efficient to just do it in the trivial way since the overhead of solving the bipartite matching subproblems is not worth the more eager cutting of branches. The trivial branch cutting can be improved by selecting the next edge that is going to be oriented following some heuristic that makes it more likely for branches to be cut off earlier. For example, we can orient first the edges that whose orientation is forced (because one of the endpoints has already indegree $f(v)-1$) and then prioritize the ones whose both endpoints have indegrees close to $f(v)$.

Implementing the above improvements already makes a very substantial impact in the execution time of the algorithm...

finish Alon-Tarsi

Algorithm 5: Optimized Alon-Tarsi.

Further improvements are possible, but we do not describe them here since this is already the implementation we have actually used for this project. For a more detailed analysis of techniques for the efficient implementation of the Alon-Tarsi method, see [**dvorakefficientalontarsi**].

Remember that the Alon-Tarsi method does not always detect colorable graphs successfully. For example, the graph in **??** is not recognized by Alon-Tarsi as a colorable graph. In addition, for bigger graphs Alon-Tarsi is significantly slower than the search for fixed small induced subgraphs. So it is still advisable to first check for the small reducible configurations we found above before running Alon-Tarsi.

fix reference to reducible configuration figure

4.4 Recursive Colorability Testing

Recursive Colorability Testing

Algorithm 6: Recursive Colorability Testing.

```
function containsColorableSubgraph(G)
    if G is empty then
        return false;
    end
    if alonTarsi(G) then
        return true;
    end
    H ← minimalNonColorable(G);
    return containsColorableSubgraph(G \ H);
end
function minimalNonColorable(G)
    for v ∈ V(G) do
        if not alonTarsi(removeVertex(G,v)) then
            return minimalNonColorable(removeVertex(G,v));
        end
    end
    return G;
end
```

4.5 Criticality Verification

Determine whether Criticality Verification section should be written and do so if necessary

Chapter 5

Approaches to the Two Precolored Triangles Theorem

5.1 Canvas Strangulation

5.2 The Forbidden 3-3 Reduction

5.3 Criticality Strength

Chapter 6

Results and Further Study

fix bibliography (not arxiv version, extraneous urls, etc

Bibliography

[1] Noga Alon. "Combinatorial nullstellensatz". In: *Combinatorics, Probability and Computing* 8.1-2 (1999), pp. 7–29.

[2] Noga Alon and Michael Tarsi. "Colorings and orientations of graphs". In: *Combinatorica* 12 (1992), pp. 125–134.

[3] T. Gallai. "Kritische Graphen I, II". In: *Publ. Math. Inst. Hungar. Acad. Sci.* 8 (1963), 165–192, 373–395.

[4] P. J. Heawood. "Map colour theorem". In: *Quarterly Journal of Mathematics* 24 (1890), pp. 332–338.

[5] J. P. Hutchinson. "On list-coloring extendable outerplanar graphs". In: *ARS MATHEMATICA CONTEMPORANEA* 5.1 (2012), pp. 175–188.

[6] Luke Postle. "5-list-coloring graphs on surfaces". PhD thesis. Georgia Institute of Technology, 2012.

[7] Luke Postle and Robin Thomas. "Five-list-coloring graphs on surfaces I. Two lists of size two in planar graphs". In: (2014). DOI: `10.48550/ARXIV.1402.1813`.

[8] Luke Postle and Robin Thomas. "Five-list-coloring graphs on surfaces II. A linear bound for critical graphs in a disk". In: (2015). DOI: `10.48550/ARXIV.1505.05927`.

[9] Luke Postle and Robin Thomas. "Five-list-coloring graphs on surfaces III. One list of size one and one list of size two". In: *Journal of Combinatorial Theory, Series B* 128 (2018), pp. 1–16. ISSN: 0095-8956. DOI: `https://doi.org/10.1016/j.jctb.2017.06.004`.

[10] C. Thomassen. "Every Planar Graph Is 5-Choosable". In: *Journal of Combinatorial Theory, Series B* 62.1 (1994), pp. 180–181. ISSN: 0095-8956. DOI: `https://doi.org/10.1006/jctb.1994.1062`.

[11] C. Thomassen. "Five-Coloring Graphs on the Torus". In: *Journal of Combinatorial Theory, Series B* 62.1 (1994), pp. 11–33. ISSN: 0095-8956. DOI: `https://doi.org/10.1006/jctb.1994.1052`.

[12] Carsten Thomassen. "Color-Critical Graphs on a Fixed Surface". In: *Journal of Combinatorial Theory, Series B* 70.1 (1997), pp. 67–100. ISSN: 0095-8956. DOI: `https://doi.org/10.1006/jctb.1996.1722`.

[13] Carsten Thomassen. "Exponentially many 5-list-colorings of planar graphs". In: *Journal of Combinatorial Theory, Series B* 97.4 (2007), pp. 571–583. ISSN: 0095-8956. DOI: `https://doi.org/10.1016/j.jctb.2006.09.002`.

[14] Carsten Thomassen. "Tilings of the Torus and the Klein Bottle and Vertex-Transitive Graphs on a Fixed Surface". In: *Transactions of the American Mathematical Society* 323.2 (1991), pp. 605–635. ISSN: 00029947.