Contents

1 Introduction (6, 8, 15, 20 september)

1.1 Basic definitions of list-criticality

We start discussing the various definitions of criticality.

Definition 1. *G is* critical (critical for k-list-coloring or $(k+1)$-list-critical) *if there exists a k-list-assignment L such that G is not L-colorable, but every proper subgraph of G can be colored with any k-list-assignment.*

We will usually work with $k = 5$, and we will usually just say *critical* when the value of k and the intention of referring to list-coloring is clear. This first definition is the one that would be most analogous (in my opinion) to the definition of critical graphs with respect to regular coloring, but we have this second definition:

Definition 2. *G is* critical *if there exists a k-list-assignment L such that G is not L-colorable but every proper subgraph of G is not L-colorable.*

This is the definition we will be working with, since it turns out to be easier to work with a fixed list assignment. Clearly, the first definition implies the second but the second does not imply the first; the graphs satisfying the first definition are sometimes called *minimal k-list-critical* because of their characterization as 5-list-critical graphs (under the second definition) with no other k-list-critical graphs as subgraphs (see [5]).

We will also use the term L-critical. The second definition is equivalent to saying that there exists a k-list-assignment L such that G is L-critical.

Definition 3. *A graph G is L-*critical *if G is not L-colorable, but every proper subgraph of G is.*

One more definition that we will be using is that of a graph critical with respect to a subgraph:

Definition 4. *Let G be a graph, T be a subgraph of G, L be a list assignment for G and ϕ be a L-coloring of T. We say that ϕ* extends to G *if there exists an L-coloring ψ of G such that $\phi(v) = \psi(v)$ for all vertices v of T. We say G is ϕ-critical if ϕ extends to every proper subgraph of G containig T but not to G.*

*The pair (G, L) is T-*critical *if for every proper subgraph G' of G which contains T as a subgraph 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.*

T-criticality can be understood as: if we denote by $\mathcal{P}_T(G)$ the set of L-colorings of T which extend to G, then G is T-critical if this set grows when any edge or vertex is removed. For our purposes, we will set T to be the outer cycle of our plane graphs, which brings us to the context of *critical canvases* as discussed in [4]:

Definition 5 ([4])**.** *We say the triple (G, C, L) is a* canvas *if G is a 2-connected plane graph, C is its outer cycle, and L is a list assignment for G, $|L(v)| \geq 5$ for all $v \in V(G) - V(C)$ and there exists an L-coloring of C. We say a canvas (G, C, L) is* critical *if G is C-critical with respect to L.*

However, we note that in the definition of T-criticality used in [4], they require that $G \neq T$; i. e. they disallow *trivially T-critical* graphs in which the criticality comes from the nonexistence of proper subgraphs containing T. For our purposes of recursively constructing critical canvases, it will be useful to consider the empty cycles as "base cases", so we will not disallow this. However, this might mean that the statements theorems cited from [4] and other references might have to be suitably modified.

The restriction to 2-connected graphs comes from this lemma, also from [4]:

Lemma 1. *If a plane graph G is C-critical, where C is its outer cycle, then G is 2-connected.*

1.2 Criticality and list-criticality on the torus

Now we will be talking about critical graphs for the usual vertex coloring (not list-coloring) for a while.

Graph coloring in surfaces is a well-studied topic. Since Heawood generalized the four color proble proved with a simple proof an upper bound on the chromatic number for graphs on any (compact) surface except the sphere (and the bound turned out to be tight for every surface except the Klein bottle), there has been much work in the area. One of the most recent and strinking contributions has been the realization that in every surface most graphs are 5-colorable, in the following sense:

Theorem 1 (Thomassen 1997)**.** *For any surface Σ, there is a finite number of 6-critical graphs which embed on it.*

So there is only a finite number of "obstructions" preventing the graph from being 5-colorable. That means also that it is possible to decide in polynomial time (in fact, linear time) whether a graph on a surface if 5-colorable. But to do that, you need to find the list of obstructions first.

Thomassen found the concrete list of obstructions for the torus:

Theorem 2 ([6])**.** *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 graph obtained by applying* Hajos' *construction to a pair of K_4.*
- 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.

Our goal is to produce similar results for list-coloring. It is also true that there is a finite number of critical graphs:

Theorem 3 (Postle, Thomas)**.** *For any surface Σ, there is a finite number of 6-list-critical graphs which embed on it.*

Given that 6-critical graphs are also 6-list-critical, the list for the torus must contain the previous four graphs too. We conjecture that there are no more graphs.

Conjecture 1. *A graph G embeddable on the torus is 5-list-colorable if and only if it does not contain the following subgraphs: $K_6, C_3 + C_5, K_2 + H_7, T_{11}$.*

How can we start trying to prove this? It is probably easier to work in the plane graph setting than with graphs embedded in the torus. If our embedded graph has two non homotopically equivalent non-contractible cycles, we can "cut the torus through the cycles" to obtain a plane graph, in which each of the vertices of the cycles appears twice in the outer face. If the graph was originally critical in the torus, then we will obtain as a result a critical canvas.

If each of the two cycles is a triangle, then the resulting canvas will have 12 vertices in the outer face. So it may be a good idea to generate all critical canvases with 12 vertices in the outer face and analyze each of them to see if they could possibly come from a critical graph embedded in a torus. Hopefully, we won't obtain any graph other than the ones listed before. And for graphs which have longer non homotopically equivalent cycles, we may have to use some stronger theoretical results to prove that they can't be critical.

1.3 Chord or Tripod Theorem and generation of critical canvases

How to generate all critical canvases with 12 vertices in the outer face? In [4] it is proven that there is a linear bound of the number of total vertices with respect to the vertices in the outer face, so the total number of such canvases is finite, but still possibly quadratically exponential. However, also in [4] is proven a structural result about critical canvases that will be very useful:

Theorem 4 (Cycle Chord or Tripod Theorem [4])**.** *If (G, C, L) is a (nontrivial) critical canvas, then either*

1. *C has a chord in G.*
2. *There exists a vertex $v \in V(G) \setminus V(C)$ 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.*

Additionally, note that every "subcanvas" of a critical canvas must be critical too. This gives us an inductive/recursive way of generating all critical canvases:

Suppose we are generating critical canvases of cycle length ℓ. There are two possible cases: canvases which have a chord, and canvases which do not. For canvases which have a chord, the chord divides the canvas into two critical subcanvases of sizes $a, b \geq 3$ with $a + b = \ell + 2$, so therefore $a, b \leq \ell - 1$ and then we can generate all such canvases if we have already generated all critical canvases with smaller cycle size.

For canvases which do not have a chord, we have a similar recursive situation: considering the "nonempty" face of the tripod, we have another critical subcanvas. Hence, having generated all smaller critical canvases, we can generate canvases of size ℓ placing a tripod in all possible ways and recursively filling one of the faces of the tripod with a smaller critical canvas. There is one problem, however: when the tripod has only three neighbors which are three adjacent vertices, the cycle size of the biggest face is ℓ, so we can't recurse to a smaller size of canvases and instead have to use canvases of the same size that we are generating now.

All in all, we have the following algorithm for generating critical canvases of size ℓ:

1. Generate all possible canvases with a chord iterating a from 3 to $\ell - 1$ and fusing together two canvases of size a and $b = \ell + 2 - a$ in all possible orientations. Put all the canvases that are critical in a queue.
2. Generate all possible canvases with a tripod with biggest face of size at most $\ell - 1$. Again, put those canvases that turn out to be critical in a queue.
3. While the queue is not empty, dequeue the first canvas, add a tripod with three consecutive neighbours to it in all the possible ways, and enqueue the canvases that turn out to be critical.

The bound on the size of critical canvases from [4] ensures that this algorithm will halt at some point. But it is necessary to test whether the canvases are actually critical or not. Doing this is unfeasible for even slightly large canvases (list-coloring is complete for the second level of the polynomial hierarchy), so what we will do instead is to test some property Π which implies criticality, but is easier to verify.

1.4 Possible tests for the Π property

First, we have the obvious observation:

Observation 1. *In a critical canvas, every interior vertex has degree at least* 5*.*

This allows us to reject a bunch of graphs with tripods with only 3 or 4 neighbours at the beginning, but we will need stronger properties. Recall Brooks' theorem from usual vertex coloring:

Theorem 5 (Brooks)**.** *Let G be a connected graph with maximum degree Δ. Then $\chi(G) \leq \Delta$ unless G is a complete graph or an odd cycle.*

There is an analogue for list coloring too:

Definition 6. *A list assignment L for G is said to be a* degree assignment *if* $\forall v\ |L(v)| \geq \deg v$.

Theorem 6 (Erdos, Rubin, Taylor 1979)**.** *Let G be a* ***2-connected*** *graph which is neither a clique nor an odd cycle. Then G is L-colorable for every degree assignment L.*

(Note that the 2-connectedness is required, by considering e.g. an odd path).

How is this useful? Well, consider a 2-connected induced subgraph G' of interior vertices of a canvas $T = (G, C, L)$ which is not a clique or an odd cycle, and suppose all the other vertices of G have been precolored. Since the vertices of G' may have some precolored neighbors, we can only suppose that the lists of the vertices of G' have the sizes of the original size of the list minus the "exterior degree" of the vertex, that is, the number of neighbours in $G \setminus G'$. And if those sizes are at least the "interior degree", then G' will always be always be colorable no matter which is the precoloring outside, so T can not be a critical canvas. In other words, we cannot have a 2-connected subgraph G' of interior vertices with $|L(v)| \geq \deg v$ for all vertices $v \in V(G')$ in a critical canvas unless G' is a clique or an odd cycle.

In our case, for all interior vertices we have $|L(v)| = 5$ and we already established that the minimum degree for interior vertices is 5, so the only candidates to be part of such a subgraph G' are the vertices with degree precisely 5. So what we do is construct the subgraph made of interior vertices with degree 5, compute its biconnected components and check that each of these is a clique or an odd cycle.

This is a good property, but it is somewhat limited since it only works with vertices of degree 5. We can find examples for which allowing greater exterior degrees (i.e. shorter lists in the subgraph) still makes the graph be always colorable.

Observation 2. *Consider the graph G with $V(G) = \{u, v, w, x\}$ and $E(G) = \{uv, vw, wx, xu, uw\}$. Then G is always L-colorable for any L with list sizes $L(u) = 3, L(v) = 2, L(w) = 2, L(x) = 2$. (This corresponds to vertex w having degree 6 in the original graph).*

We can collect a list of these forbidden subgraphs and list sizes and manually check if the graph contains them in order to test a property stronger than what the theorem gives us. This allows for many possibilities, because the forbidden subgraphs don't necessarily need to be always L-colorable with the given list sizes, but can preclude the whole graph from being critical in other ways. Take this example:

Observation 3. *Consider the graph G with $V(G) = \{u, v, w, x, y\}$ and $E(G) = \{uv, uw, ux, uy, vw, wx, xy\}$. Consider the list sizes $L(u) = 4, L(v) = L(w) = L(x) = L(y) = 2$. Then G is not always L-colorable with any L assignment with those list sizes, but any L-coloring with these list sizes of G without edge wx extends to G.*

Proof. We will prove that if there is a common color $c \in L(w) \cap L(x)$, then the graph can always be L-colored, so that for any assignment L for which the graph is not L-colorable, the graph remains not L-colorable when removing the edge wx (since w and x can not be the same color).

If there exists such a c, then there exists a $c' \in L(u)$ such that $c' \notin L(w), c' \notin L(x)$. Now, if $c' \notin L(v)$ or $c' \notin L(y)$, then we can assign c' to u and the remaining graph is a path with lists sizes $1, 2, 2, 2$, which can always be colored. If $c' \in L(v) \cap L(y)$, then assigning c' to v and y results in a triangle with lists sizes $2, 3, 2$, which can always be colored. □

However, we need not go so far and it will probably be sufficient to forbid subgraphs which can always be colored with some prescribed list sizes. Instead of trying to find these subgraphs by hand, there is a more systematic way of checking whether a graph with given list sizes can always be L-colored. For doing so, we will need the Combinatorial Nullstellensatz technique from Noga Alon.

1.5 Combinatorial Nullstellensatz and applications

This section will be mostly sourced from [1].

Theorem 7 (Combinatorial Nullstellensatz)**.** *Let $p(x_1, \ldots, x_n) \in \mathbb{K}[x_1, \ldots, x_n]$ be a nonzero polynomial and let $t_1, \ldots, t_n$ be nonnegative integers such that the degree of x_i in p is at most t_i. Let $S_1, \ldots, S_n$ be subsets of $\mathbb{K}$ such that $|S_i| = t_i + 1$. Then there exist $a_1 \in S_1, \ldots, a_n \in S_n$ such that*

$$p(a_1, \ldots, a_n) \neq 0.$$

We will use the graph polynomial associated with a directed graph G:

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

This polynomial is related to coloring since a coloring ϕ is proper iff $p_G(\phi(v_1), \ldots, \phi(v_n)) \neq 0$. So we can see how the combinatorial nullstellensatz is going to be useful here. The important result is this one:

Theorem 8 (Alon, Tarsi)**.** *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 the coefficient of p_G at $\prod_{i=1}^n x_i^{d^+(v_i)}$ is non-zero, then G has an L-coloring.*

This coefficient has a combinatorial interpretation:

Theorem 9. *The $\prod_{i=1}^n x_i^{d^+(v_i)}$ coefficient is the difference between even and odd spanning eulerian subgraphs of G. (*Even *and* odd *refer to the number of edges).*

(Here is a brief detour from what concerns us). Hence, since bipartite graphs have no odd spanning eulerian subgraphs (the number of edges in each direction on the bipartition is the same), any orientation gives a prescribed set of list sizes so that the graph is always L-colorable. Additionally, we have the following lemma:

Lemma 2. *A graph G has an orientation with maximum indegree d if and only if $|E(H)| \leq d|V(H)|$ for every induced subgraph H.*

Since bipartite planar graphs have at most $2n$ edges, we can conclude the following:

Theorem 10. *Every planar bipartite graph can be 3-list-colored.*

This theorem can also be proved by a more combinatorial approach without the nullstellensatz (see [1]).

Cute historical note: this theorem, along with Thomassen's theorem and the Four Color Theorem, leads to conjecture that every planar graph has list chromatic number at most its chromatic number plus one. However, Mirzakhani provided a counterexample to this conjecture (I did a video about that!).

How can these results help us? They provide a systematic way to find gadgets to avoid on our critical canvases. However, it will be more practical to check whether the entire graph induced by the interior vertices of our canvas (and with list sizes set by the number of neighbours in the outer face) can always be colored according to the preceding theorems. To do so, we will use the odd and even eulerian orientations characterization (modified slightly from the eulerian subgraph phrasing; we consider orientations having the desired indegrees and count the number of flipped edges compared to some fixed orientation). This is because generating those orientations by a pruned exhaustive search will be more efficient than computing the entire graph polynomial. There is a way to turn this orientation problem into a variant of matching in a general graph, making it polynomial time, but my supervisor thinks the exhaustive search will be more efficient in practice.

2 Implementation of the search program (8 september - 3 october)

Here we briefly comment some aspects about the implementation of the search program for critical canvases.

2.1 Basic data structures to store canvases

The program we have written has two main files: `search.cpp` and `PlaneGraph.cpp`. search.cpp contains the main algorithm at a higher level of abstraction for recursively generating candidates for critical canvases out of smaller critical canvases, while `PlaneGraph.cpp` contains the detailed, low-level implementation of the specific algorithms and data structures used for all the operations.

The most basic aspect of the program is how to store the canvases themselves. We use the struct named `PlaneGraph`, which contains the graph represented in an adjacency list format, where the neighbors in the adjacency list are stored in clockwise order with respect to the embedding, and we also distinguish which vertices are part of the outer face. One problem we face is that, in our generation procedure, we will generate essentially the same canvases multiple times, and we only need to store one copy of each distinct canvas in order to prevent the number of canvases to grow too much. That is, we need to be able to tell when two canvases are isomorphic. Fortunately, since we are working with planar graphs with a fixed embedding, checking isomorphism is easy: we just need to traverse the graph depth-first following a consistent orientation, and the lexicographically minimum of such traversals among all the possible starting vertices and edges can serve as a canonical representation of the graph.

2.2 Recursive Alon-Tarsi test

After implementing all the tests described above, we find that our program is still not efficient enough to generate all critical canvases with up to 12 vertices. The Alon-Tarsi test is the one that is most helpful to eliminate candidates, and it works especially well with canvases with a larger number of vertices (which are the main ones we are interested in discarding), but taking a look at the generated canvases makes us realize that the fact that we only test the subgraph induced by all the interior vertices of the graph instead of checking smaller subgraphs too makes the test miss some non-critical canvases which have a large reducible (always-colorable) subgraph, but which have a small number of interior vertices that prevent the interior from being always colorable and therefore the canvas passes the Alon-Tarsi test.

However, the test is already inefficient enough for checking one subgraph per canvas, and checking *all* subgraphs would be prohibitively inefficient. We can follow a smarter strategy: if a graph passes the Alon-Tarsi test, we find a *minimal* subgraph that passes the Alon-Tarsi test; i.e. which can not be shown to be always colorable with the prescribed list sizes but if we remove any vertex then it becomes colorable. We do this by trying to remove each of the interior vertices, applying the Alon-Tarsi test and recursively finding the minimal resulting subgraph if it comes out positive. Once we have this minimal subgraph, we arbitrarily choose one of the vertices v of the subgraph and we *precolor* it, that is, now this vertex is, like the vertices of the outer face, assigned an arbitrary color and we are interested in whether the rest of the graph can be colored no matter which color is chosen for this vertex. Since we don't have implemented the capacity to consider arbitrary vertices to be precolored in our program, what we will do instead will be to recursively consider the subcanvases for which v forms part of the outer face, and recursively apply this *Recursive Alon-Tarsi test* to them. We stop when we reach some subcanvas which doesn't pass the Alon-Tarsi test, in which case we know the complete canvas can not be critical, or if we reach an empty canvas, in which case we backtrack.

This strategy allows us to identify more non-critical canvases because, if the situation is as described above (large reducible interior subcanvas, some interior vertices prevent the whole interior graph from being reducible), then by identifying the minimal subgraphs which are not reducible we will precolor those vertices and then find the large reducible subgraph. Do note that, since performing the Alon-Tarsi test multiple times is very inefficient, we need to memoize the results of the tests for all canvases.

After doing this, our program is finally efficient enough to generate all candidates for critical canvases with outer face of length 12.

2.3 Canvas criticality verifier

3 Two precolored triangles

In Postle's thesis, the following result is proven:

Theorem 11. *Let G a planar graph, L a list assignment with lists of size 5 and T_1, T_2 be two triangles on it with distance between them at least 14 (the distance between them is the minimum distance between any pair of vertices). Then any valid precoloring of $T_1 \cup T_2$ extends to the whole graph.*

Our goal now will be to improve the bound of 14 and find the tightest possible bound.

We want to generate all $(T_1 \cup T_2)$-critical graphs, where T_1 and T_2 are triangles at distance d, for different values of d. Our hope is that, for some small value of d such as 4 or 5, we will find no critical graphs, so we will have found the tightest bound possible in the above result by Postle. (Of course, after that we will have to think how to prove the bound).

3.1 Two precolored triangles from critical canvases

One way to generate all $(T_1 \cup T_2)$-critical graphs is as follows: consider a $(T_1 \cup T_2)$-critical graph G which has T_1 and T_2 joined by a path of length d. Now, duplicate all $d-1$ vertices in the interior of the path and let C be the resulting cycle of length $2d+6$ formed by the duplicated vertices of the path and the two triangles. Let G' be the resulting graph.

Theorem 12. *G' is C-critical.*

Proof. Let e' be an edge not in C, consider the corresponding edge e from G: by $(T_1 \cup T_2)$-criticality, there is a coloring of $(T_1 \cup T_2)$ which extends to $G\ e$ but not to G. Let ϕ_P be the extension of that coloring to the path between T_1 and T_2 and let ϕ_C be the precoloring of C in which the corresponding duplicated vertices get the same color. Now, note that ϕ_C can be extended to $G\ e'$ but not to G, since if it did we could recover an extension of the original precoloring of the two triangles. □

Hence, in order to enumerate all $(T_1 \cup T_2)$-critical triangles with triangles at distance at most d, we can generate all critical canvases with outer face of size $2d+6$, and for each canvas find all the ways to merge vertices in order to obtain the graph with two triangles. Of course, this approach will generate many graphs which are not critical - we can discard some of them using Alon-Tarsi or other tests.

However, we will also try a different approach for enumerating the critical graphs with two precolored triangles. This different approach will make it easier to translate the results of our computations into a proof.

3.2 Generating critical graphs with a precolored path

As a prerequisite, we will see how to generate all planar graphs with interior vertices with lists of length 5 and outer vertices with lists of length 3 critical with respect to a precolored path P on the boundary. Recall that, by Thomassen's theorem, there are no such critical graphs for paths of length 1.

We will be using the following result from Postle:

Theorem 13 ([3])**.** *Let G be a planar graph and L a list assignment with lists of size 5 for interior vertices and lists of size 3 for exterior vertices except for two of those, which have lists of size 2. Then G is L-colorable.*

Using this, we can identify 3 possible cases in which a P-critical graph might fall into:

1. The graph has a chord inside P, i.e. there is an edge between two vertices of P which is not part of the path. Then, the graph can be partitioned into two smaller graphs critical with respect to some smaller path.

2. The graph has a chord form P to some outer vertex not in P: then, similarly, the graph can be decomposed into two smaller graphs critical with respect to shorter paths.

3. Otherwise, here is the more interesting case: The key idea is that, if all interior vertices have at most two neighbors in P, then by removing P and removing the corresponding colors in the list we obtain a graph which satisfies the hypothesis Postle's result (all exterior vertices have list size at least 3, save for the two ones which are adjacent to the ends of the paths which have list size at least 2), and therefore is always colorable, so the graph is not critical. Hence, we must have at least one *tripod*, or a graph with at least three neighbors in P, and furthermore we can take such tripod to divide the graph into faces such that only one of them is nonempty. The graph bounded by the nonempty face must be critical with respect to a shorter or equal path: note that here the path (and the outer cycle) might have equal length so that, if we're generating the critical graphs by length of outer cycle, we must use a queue like in the critical canvas scenario.

With this recursive characterization, we can generate P-critical graphs in a very similar way to which we generated critical canvases. Having those graphs generated will be useful for our goals.

3.3 Generating critical graphs with two precolored paths

We still need to generate another type of critical graph before we can generate our triangles. Now, we will be generating graphs critical with respect to *two* precolored paths of length 1. This will be a more difficult endeavor, and we will have to change our strategy a bit.

Inspired by the argument developed in paper [2], we will work in a different setting: instead of requiring only that all vertices in the outer face have list size at least 3, we will also require that no two adjacent vertices have list size exactly 3 (i.e. at least one of them has size at least 4). In this setting, it is proved in [2] that there are only 2 critical graphs with respect to a precolored path of length 2.

Let's see now how to generate all graphs critical with respect to two precolored paths of length at most 1 in this setting. We again have three cases, the first two of which are easy:

1. The outer face is not bounded by an induced cycle; i. e. the graph has an articulation point or a chord. Then, in the case of the articulation point, the graph can be divided into two graphs critical wrt a path of length 1 and a path of length 0, and in the case of a chord the graph can be divided wrt into two critical graph wrt paths of length 1.

2. The two precolored paths are *near*, i. e. the distance between them is not very large. Then the graph must be also critical with respect to the entire path joining the two vertices, and we can use the previouss section where we enumerated the graphs critical with respect one precolored path.

The third case, when the two preocolored paths are far apart, is more complex. The strategy we will follow will be to analyze the different cases for the list sizes of the vertices which are near one of the paths, similarly to what is done in [2]. For each of the cases, smartly choosing which vertices to precolor leaves us with a smaller graph for which the precoloring of the graph does not extend. However, we need to be careful here: this graph might not be critical itself, it just has to *contain as a subgraph* another critical graph. Therefore, this case requires a more subtle recursive generation procedure.

3.4 Generating critical graphs with a precolored path and a precolored triangle

References

[1] Z. Dvorak. "Lesson on List Coloring". In: (). URL: https://iuuk.mff.cuni.cz/~rakdver/kgiii/lesson14-8.pdf.

[2] Zdeněk Dvořák, Bernard Lidický, and Bojan Mohar. "5-choosability of graphs with crossings far apart". In: *Journal of Combinatorial Theory, Series B* 123 (2017), pp. 54–96. DOI: 10.1016/j.jctb.2016.11.004. URL: https://doi.org/10.1016%2Fj.jctb.2016.11.004.

[3] 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. URL: https://arxiv.org/abs/1402.1813.

[4] 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. URL: https://arxiv.org/abs/1505.05927.

[5] Michael Stiebitz, Zsolt Tuza, and Margit Voigt. "On list critical graphs". In: *Discrete Mathematics* 309.15 (2009). Cycles and Colourings, pp. 4931–4941. ISSN: 0012-365X. DOI: https://doi.org/10.1016/j.disc.2008.05.021. URL: https://www.sciencedirect.com/science/article/pii/S0012365X08003506.

[6] 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. URL: https://www.sciencedirect.com/science/article/pii/S0095895684710525.