Graph Choosability via SAT: Beyond the Nullstellensatz

Markus Kirchweger[1], Tomáš Peitl[1], David Seka[1], Stefan Szeider[1]

[1]Algorithms and Complexity Group, TU Wien, Austria
{mk,peitl,seka,sz}@ac.tuwien.ac.at

Abstract

List coloring extends graph coloring by assigning each vertex a list of allowed colors. A graph is k-choosable if it can be properly colored for any choice of lists with k colors each. Deciding k-choosability is Π_p^2-complete, bipartite graphs have unbounded list chromatic number, and planar graphs (famously 4-colorable) are all 5-choosable but not all 4-choosable.

To search for graphs of given choosability, we extend SAT Modulo Symmetries (SMS) with custom propagators for list coloring pruning techniques and propose a quantified Boolean (QBF) encoding for choosability. We employ a hybrid approach: pen-and-paper reasoning to optimize our formulas followed by automated case distinction by QBF solvers and SMS.

Our methods yield two significant results: (1) a 27-vertex planar graph that is 4-choosable yet cannot be proven so using the combinatorial Nullstellensatz widely applied in previous work (we show this is a smallest graph with that property), and (2) the smallest graph exhibiting a gap between chromatic and list chromatic numbers for chromatic number 3.

1 Introduction

In graph coloring, the task is to properly color the vertices of a graph with k colors, i.e., so that adjacent vertices have different colors. List coloring is a variant in which each vertex has an explicit list of available colors.

A bipartite graph is 2-colorable, i.e., list-colorable when each vertex has the same list of two colors. Can any bipartite graph be colored whenever lists of size two are assigned to the vertices, even if (some of) those lists are different? No: Figure 1 shows a bipartite graph along with a *bad list assignment*: an assignment of lists that does not admit any proper coloring. For other graphs, no such bad list assignment with lists of size k may exist: in that case we say that the graph is *k-choosable*. Deciding, whether a graph is k-choosable is Π_p^2-complete (Erdős, Rubin, and Taylor 1979).

In this paper, we use SAT-based computational methods to tackle conjectures and open questions around choosability of graphs. Our main interest lies in finding graphs which exhibit a gap between the *chromatic* and *list chromatic* numbers: the smallest number of colors needed to color properly,

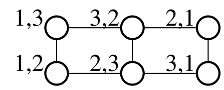


Figure 1: The smallest non-2-choosable bipartite graph with a bad list assignment. The middle vertices must be colored 2 and 3; either way leaves only 1 for one of the side pairs.

and the smallest list size to grant choosability. By definition, the list chromatic number is no smaller than the chromatic number (all-lists-equal is a bad list assignment), but the list chromatic number is not bounded by any function of the chromatic number: bipartite graphs have unbounded list chromatic number. Of particular interest to us are the smallest *gap* graphs for $k \in \mathbb{N}$: k-colorable, but not k-choosable.

The smallest gap graphs for $k = 2$ are known (one is in Figure 1). Our first set of results revolves around the smallest gap graphs for $k = 3$. We find the smallest graph in terms of both the number of vertices and edges that is 3-colorable but not 3-choosable (Figures 2 and 3). In order to achieve this, we employ *SAT Modulo Symmetries* (*SMS*; Kirchweger and Szeider 2024) for isomorph-free graph generation, in particular the recently introduced support for *quantified Boolean formulas (QBF)* in SMS (Janota et al. 2025). We develop a QBF encoding of choosability, and implement custom propagators for advanced choosability techniques based on the *combinatorial Nullstellensatz* (Alon and Tarsi 1992). We emphasize this: choosability can be encoded into QBF relatively easily, but without custom propagators the encoding is not practically solvable. Custom propagators can easily be integrated into SAT solvers thanks to IPASIR-UP (Fazekas et al. 2024), an extension to the well-known IPASIR interface, supported in recent versions of CaDiCaL (Biere et al. 2024) and some QBF solvers (Janota et al. 2025).

Next, we consider planar graphs. The famous four-color theorem asserts that each planar graph is 4-colorable, but it is also known that some are not 4-choosable: the smallest known such graph (63 vertices) is due to Mirzakhani (1996). We were unable to find a smaller planar graph that is not 4-choosable, but we discovered a small planar graph (27 vertices, Figure 6) that, while 4-choosable, cannot be proven 4-choosable using the combinatorial Nullstellensatz, a sufficient condition widely used in previous computational work on choosability of planar graphs (Nelsen 2019).

We prove our graph 4-choosable with a hybrid method that uses pen-and-paper arguments to compress the QBF encoding, followed by a QBF solver to complete the case distinction. This underscores the importance of hybrid SAT and QBF-based approaches: in order to obtain practical results, it seems necessary to produce complicated encodings with involved theoretical arguments, and to use custom propagators: a naive encoding of choosability would hardly scale. We also speed up Nelsen's method with new ideas and extend his result to all 26-vertex planar graphs, establishing that the graph in Figure 6 is smallest possible.

After preliminaries, we review list coloring in Section 3. We present our encoding of choosability in Section 4, the improvements necessary to obtain the smallest gap graphs for $k = 3$ in Section 5, and the planar graph that cannot be proved choosable via Nullstellensatz alone in Section 6.

2 Preliminaries

For a positive integer n, we write $[n] = \{ 1, 2, \ldots, n \}$. Below we review basics on SAT and graph theory.

SAT & QBF

Quantified Boolean formulas (QBF) generalize propositional logic with quantification. We consider formulas in *prenex* form, where all quantifiers are in front in the quantifier *prefix*, and the rest—the *matrix*—is a propositional formula. The matrix is not required to be in CNF, it can be an arbitrary circuit, and can contain *free variables*, e.g.

$$\exists x \exists y \forall z \big((x \wedge \neg y \wedge \neg a) \vee (z \wedge a) \big).$$

For actual solving, the formula is made *closed* by quantifying each free variable as leftmost existential in the prefix. A closed prenex QBF is either true or false, defined by recursive quantifier expansion in the usual way.

An *assignment* to a set of variables V is a mapping $\tau : V \to \{0, 1\}$, extended to negations and arbitrary propositional formulas in the natural way. For a QBF Φ and an assignment $\alpha : \text{var}(\Phi) \supseteq V \to \{ 0, 1 \}$, $\Phi[\alpha]$ is the QBF in which V is replaced with the respective truth values, any bound variables in V are removed from the prefix, and the matrix is simplified accordingly. An assignment α to the free variables of Φ is a *satisfying assignment* or a *model* of Φ if $\Phi[\alpha]$ is true. A QBF Φ is *satisfiable* if there is a (*satisfying*) assignment α to the free variables such that $\Phi[\alpha]$ is true; otherwise it is unsatisfiable.

We store QBFs in the standard *QCIR format* (Jordan, Klieber, and Seidl 2016).

Graphs

We consider simple, undirected and unweighted graphs without parallel edges or self-loops. A *graph* G consists of a set $V(G)$ of vertices and a set $E(G) \subseteq \binom{V(G)}{2}$ of edges; we denote the edge between vertices $u, v \in V(G)$ by uv or equivalently vu. We write $\mathcal{G}_n$ to denote the class of all graphs with $V(G) = [n]$. The *adjacency matrix* of a graph $G \in \mathcal{G}_n$, denoted by A_G, is the $n \times n$ matrix where the element at row v and column u, denoted by $A_G(v, u)$, is 1 if $vu \in E$ and 0 otherwise. For a permutation $\pi : [n] \to [n]$, $\pi(G)$ denotes the graph obtained from $G \in \mathcal{G}_n$ by the permutation π, where $V(\pi(G)) = V(G) = [n]$ and $E(\pi(G)) = \{ \pi(u)\pi(v) \mid uv \in E(G) \}$. Two graphs $G_1, G_2 \in \mathcal{G}_n$ are *isomorphic* if there is a permutation $\pi : [n] \to [n]$ such that $\pi(G_1) = G_2$; in this case G_2 is an *isomorphic copy* of G_1. For $W \subseteq V(G)$, $G - W$ denotes the graph obtained from G by deleting the vertices W and any edges incident to them.

SAT Modulo Symmetries (SMS)

Modern propositional satisfiability (SAT) solvers are primarily based on *conflict-driven clause learning* (CDCL, Fichte et al. 2023; Marques-Silva, Lynce, and Malik 2021). SMS is a framework that augments a CDCL SAT solver[1] with a custom propagator, allowing the SAT solver to search *modulo isomorphisms* for graphs with a given number n of vertices and which satisfy constraints specified by a propositional formula. The SMS propagator (the *minimality check*) checks whether the currently explored branch of the search tree contains a *canonical graph*: a distinguished member of its isomorphism class. In SMS this is the isomorphic copy with lexicographically minimal adjacency matrix, where matrices are compared row-wise: the vector of $\left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ is $\left(\begin{smallmatrix} 0 & 1 & 0 & 0 \end{smallmatrix}\right)$. The minimality check is called by solver when unit propagation reaches fixpoint, and can trigger an additional conflict and learn a *symmetry-breaking clause*, which excludes non-canonical graphs.

The minimality check is only one of a set of external propagators implemented in SMS. These propagators can be used as a substitute for constraints that would be difficult to encode in propositional logic. A typical such example is non-3-colorability, which cannot be encoded in a polynomial-size propositional formula unless $\mathrm{NP} = \mathrm{coNP}$.

SMS also supports QBF solving, via both an integrated 2-QBF solver (limited to one quantifier alternation), as well as integration with the general-purpose QBF solvers CQesto, Qfun, and Qute (Janota et al. 2025).

This paper shows how custom propagators in SMS can substantially reduce the running time of choosability checking, underscoring the importance of a flexible framework that supports the integration of domain-specific propagators.

3 Graph Coloring and Choosability

In this section we introduce list coloring and choosability, following the notation of West (2020).

A *proper k-coloring* of a graph G is a mapping $\phi : V(G) \to [k]$ such that $\phi(u) \neq \phi(v)$ for all $uv \in E(G)$. The *chromatic number* $\chi(G)$ is the least k for which a proper k-coloring of G exists. Sometimes colors may be forbidden for certain vertices, giving rise to list coloring and choosability.

For a graph G, a *list assignment* L is a mapping that assigns to each vertex $v \in V(G)$ a finite set $L(v)$ of colors allowed at v. A *proper L-coloring* is a mapping $\phi : V(G) \to \bigcup_v L(v)$ such that $\phi(v) \in L(v)$ and $\phi(u) \neq \phi(v)$ for $uv \in E(G)$. A list assignment L is *bad* if G has no L-coloring. For a color c, the set $A_c = \{v \in V(G) : c \in L(v)\}$

[1]At the moment, SMS uses the SAT solver CaDiCaL (Biere et al. 2024) via the IPASIR-UP interface (Fazekas et al. 2024).

of vertices that have c on their lists is called a *colorability class*. A list assignment L is identified, up to symmetry of colors, by the multiset $A_L = \{ A_c \mid c \in \bigcup_{v \in V(G)} L(v) \}$ of colorability classes. The *width* $|\bigcup_{v \in V} L(v)|$ of L is the total number of colors used on the lists of L; also the number of colorability classes $|A_L|$. For a graph G and $f : V(G) \to \mathbb{N}$, a list assignment L is *f-full* if $|L(v)| = f(v)$ for all v. A graph G is *(width-w) f-choosable* if G has a proper L-coloring for all list assignments L (of width at most w) with $|L(v)| \geq f(v)$ for all $v \in V(G)$; G is *(width-w) k-choosable* if it is (width-w) f-choosable for the constant function $f : v \mapsto k$. The *list chromatic number* $\chi_l(G)$ is the least k such that G is k-choosable.

The list assignment $L(v) = [k]$ for every vertex v shows that a k-choosable graph is also k-colorable: $\chi(G) \leq \chi_l(G)$. The converse, however, does not hold, and in fact $\chi(G)$ and $\chi_l(G)$ can be arbitrarily far apart (Erdős, Rubin, and Taylor 1979).[2] In this paper, we investigate the gap between χ and χ_l for various graph classes, in particular for small 3-colorable graphs and small *planar* graphs. A graph is planar if it can be drawn in the plane without crossings. The famous four color theorem (Appel, Haken, and Koch 1977; Robertson et al. 1997) asserts that every planar graph is 4-colorable. Thomassen (1994) proved that every planar graph is 5-choosable, and Voigt (1993) constructed a non-4-choosable planar graph with 238 vertices, improved to 63 vertices by Mirzakhani (1996). Nelsen (2019) asked for a smallest non-4-choosable planar graph, and proved that every planar graph with $\leq$ 24 vertices is 4-choosable. He achieved this by generating all non-isomorphic planar *triangulations* with `plantri` (Brinkmann and McKay 2007) and checking each for 4-choosability. Checking whether a graph is k-choosable is Π_2^P-complete (Erdős, Rubin, and Taylor 1979); Nelsen circumvented this barrier by checking a sufficient condition for 4-choosability based on an application of the *combinatorial Nullstellensatz* (Alon and Tarsi 1992) to the context of list coloring.

Theorem 1 (Combinatorial Nullstellensatz for list coloring, Alon and Tarsi 1992)**.** *Let G be a graph, and $f : V(G) \to \mathbb{N}$. The* graph polynomial *p_G is defined as*

$$p_G(\{x_v : v \in V(G)\}) = \prod_{uv \in E} (x_u - x_v).$$

If p_G contains (with a non-zero coefficient) a monomial $x_{v_1}^{d_1} \dots x_{v_n}^{d_n}$ with $d_i < f(v_i)$ for all i, then G is f-choosable.

We call a graph *k-NS-choosable* if it can be proven k-choosable by the Nullstellensatz, and *k-NS-oblivious* if the Nullstellensatz yields no result (we omit the k if it is clear from the context). A k-NS-choosable graph is k-choosable, but a k-NS-oblivious graph may or may not be k-choosable. Nelsen's result is: all planar graphs with ≤ 24 vertices are 4-NS-choosable.[3]

[2] The complete bipartite $K_{\binom{2k-1}{k},\binom{2k-1}{k}}$ is not k-choosable.

[3] Technically, the result is that all planar triangulations are 4-NS-choosable, but NS-choosability is monotone in subgraphs, and so transfers to all planar graphs. To see this, consider that if H is a subgraph of G, then $p_H | p_G$, and thus monomials in p_G have greater multi-degree than those in p_H.

4 Encoding

In this section we describe a QBF encoding of f-choosability. Our encoding expresses the existence of a bad list assignment. In order to express this in a finite formula, we first need a bound on the width (the total number of colors) in terms of the number of vertices.

Lemma 1 (Nelsen 2019, Lem 3.11)**.** *Let $n = |V(G)|$, $w = n - 1$. If G is width-w f-choosable, then G is f-choosable.*

Lemma 1 says that non-choosability can always be witnessed with a bad list assignment using up to $n-1$ colors. As we will see later, we want to constrain the encoding to even fewer than $n-1$ colors, as the difficulty of the formula grows rapidly with increasing width.

Let $E_n = \{ e_{u,v} \mid 1 \leq u < v \leq n \}$ be the set of edge variables for graphs with n vertices. Below we describe a prenex QBF $F_{n,f}^w(E_n)$ with free variables E_n whose satisfying assignments (to E_n) are in 1-to-1 correspondence with width-w non-f-choosable graphs with n vertices. With $F_{n,f}^w(E_n)$ we can search for a non-f-choosable graph with a fixed number of vertices n, or, by plugging in a concrete graph as partial assignment, test whether a graph is width-w f-choosable. Using the planarity propagator of SMS, we can extend the encoding to search for planar graphs. We additionally implemented a propagator for the combinatorial Nullstellensatz (Theorem 1), in order to filter out NS-choosable graphs and speed up the search. Let $F_{n,f}^w(E_n) = \exists L \forall C \, \mathit{full}(L,f) \wedge \mathit{overlaps}(L) \wedge \neg \mathit{col}(L,C)$,

$$\mathit{full}(L,f) = \bigwedge_{v \in [n]} \sum_{i \in [w]} l_v^i = f(v)$$

$$\mathit{overlaps}(L) = \bigwedge_{a,b \in [w]}^{a<b} \left(\bigwedge_{v \in [n]} \neg l_v^a \vee \bigwedge_{v \in [n]} \neg l_v^b \vee \bigvee_{v \in [n]} (l_v^a \wedge l_v^b) \right)$$

$$\mathit{col}(L,C) = \mathit{listcol}(L,C) \wedge \mathit{proper}(C)$$

$$\mathit{listcol}(L,C) = \bigwedge_{v \in [n]} \left(\bigvee_{i \in [w]} c_v^i \wedge \bigwedge_{i \in [w]} (\neg c_v^i \vee l_v^i) \right)$$

$$\mathit{proper}(C) = \bigwedge_{1 \leq u < v \leq n} \bigwedge_{i \in [w]} (\neg e_{u,v} \vee \neg c_u^i \vee \neg c_v^i)$$

where $L = \{ l_v^i \mid v \in [n], i \in [w] \}$ is a set of variables that encode a list assignment and $C = \{ c_v^i \mid v \in [n], i \in [w] \}$ is a set of variables that encode a coloring, whereby l_v^i is true if and only if the color i is on the list for the vertex v, and c_v^i is true if and only if the color i is selected for the vertex v. We write $\sum$ as shorthand for a cardinality constraint, which we encode as a Boolean circuit using a sequential counter (Sinz 2005). Furthermore, we add restrictions that require the colorability classes to be ordered lexicographically.

Bounding the Number of Colors (Width)

The width parameter has a significant impact on the difficulty of the associated encoding $F_{n,f}^w$ (c.f. Tables 1 and 2). The tightest possible upper bound on width is desired. The best known general bound on the width of a bad list assignment is $n-1$ from Lemma 1. Below, we discuss better bounds that can be obtained when n or k are small.

Observation 1. *In any bad list assignment of minimum width, each pair of colorability classes overlaps.*

Proof. Merging disjoint colorability classes reduces width and preserves f-fullness and badness. ☐

Lemma 2. *Let G be not k-choosable. Then there exists a bad list assignment of width w such that $\binom{w}{2} \leq \binom{k}{2} \cdot |V(G)|$.*

Proof. Consider a bad list assignment L of minimum width. By Observation 1, for each pair of colorability classes A_1, A_2 there exists a vertex v such that $v \in A_1 \cap A_2$, and there are $\binom{w}{2}$ such pairs. Call the triple (v, A_1, A_2) a witness. Let $v \in V(G)$: as $|L(v)| = k$, there are exactly $\binom{k}{2}$ pairs of colorability classes A_1, A_2 such that $v \in A_1 \cap A_2$. Thus, the number of witnesses is between $\binom{w}{2}$ and $\binom{k}{2}|V(G)|$. ☐

Lemma 3. *Let G be a graph that is not k-choosable, and such that every bad list assignment L' has width at least p. Then there must exist a bad list assignment L where for each colorability class A we have $|A| \geq \lceil \frac{p-1}{k-1} \rceil$.*

Proof. Fix a colorability class A: the number of witnesses (v, A, A') with arbitrary v and $A' \neq A$ is $|A|(k-1)$; but it is also $\geq p-1$ since A intersects with all other A'. ☐

Corollary 1. *Let G be a non-3-choosable graph with 9 vertices. Then G has a bad 3-full list assignment of width ≤ 7.*

Proof. Let L be a bad 3-full assignment of minimum width with the colorability classes $\mathcal{A}_L$, $|\mathcal{A}_L| \geq 8$. By Lemma 3, for each $A \in \mathcal{A}_L$, $|A| \geq 4$. Then $\sum_{A \in \mathcal{A}_L} |A| \geq 8 \cdot 4 = 32$ but $\sum_{A \in \mathcal{A}_L} |A| = 9 \cdot 3 = 27$ by 3-fullness. ☐

Corollary 1 reduces width for gap graphs for $k = 3$ from 8 to 7. This is important, as a lower width drastically reduces the number of potential list assignments.

5 Smallest Gap Graphs for $k = 3$

With full enumeration and the combinatorial Nullstellensatz, Nelsen found the smallest gap graphs for $k = 2$. He also searched graphs with ≤ 8 vertices: finding no gap graphs for $k = 3$, he asked for the smallest such graph (Nelsen 2019, Question 3.19). Noel, Reed, and Wu (2012) proved that if $|V(G)| \leq 2\chi(G) + 1$, then G is $\chi(G)$-choosable. Enomoto et al. (2002) showed that this bound is tight precisely when $\chi(G)$ is even and characterized such graphs and their bad list assignments. Notably, the edge-minimal gap graph for even k on $2k+2$ vertices is width-$(2k-1)$-choosable and has a bad list assignment of width $2k$. Therefore one cannot hope for a tighter general bound on width than $|V(G)| - 2$.

In this paper we find the smallest gap graphs for $k = 3$ (they have 9 vertices). Along the way we confirm (for $r = 3$) a result of Kierstead (2000) that an r-partite graph with partitions of size 3 has list chromatic number $\lceil (4r-1)/3 \rceil$.

We build on the encoding from Section 4, but the encoding alone is already hard for small graphs.

Another important aspect is the number of colors (width). The number of possible list assignments (and solving time) grows tremendously with each additional color, as shown in Tables 1 and 2. We use the results from Section 4 to bound the width as tightly as we can.

The Vetting Method

One drawback of the combinatorial Nullstellensatz is it provides only a sufficient condition: there are f-choosable graphs for which NS is inconclusive. Nelsen (2019) developed an exact "vetting method" based on NS.

Lemma 4 (Nelsen 2019)**.** *Let G be a graph, $f : V(G) \to \mathbb{N}$, $A \subseteq V(G)$, and $A^I \subseteq A$ an independent set. Let $G' := G - A^I$, and $f'(v) := f(v) - 1$ if $v \in A$ and $f'(v) = f(v)$ otherwise. Then, if G' is f'-choosable, A cannot be a colorability class in any bad list assignment of G.*

Proof. Any L'-coloring of G' for an f'-full list assignment L' can be extended to A^I with a fresh color. ☐

In practice, this tends to reduce the number of colorability classes dramatically, even if one only uses the combinatorial Nullstellensatz to determine the choosability of G'. We deploy Lemma 4 as a preprocessor for our encodings when checking choosability of a fixed graph. This provides a significant boost in performance, especially on otherwise hard instances. We use an efficient implementation of the Nullstellensatz due to Dvořák (2024).

The Vetting Method in the Context of SMS

In the graph search context no concrete graph is given, which complicates the application of Lemma 4 as a preprocessing step. However, it is possible to write a propagator based on the same principles. When all the variables defining a graph G and some colorability class A are set, the propagator checks for each maximal independent subset A^I whether G' is f'-choosable. This lets us rule out the colorability class A for the graph G. For a set of graphs and a set of colorability classes we can use a similar argument. Let $\hat{G}$ be the graph resulting from G by adding all possible edges from A^I to all other vertices. Now, we can rule out every colorability class $\hat{A}$ that is a subset of A and a superset of A^I (as $A^I \subseteq \hat{A}$ and $\hat{f}' \geq f'$), for each subgraph of $\hat{G}$.

Thus, for a fully defined graph G, colorability class A, and independent set $A^I \subseteq A$ where Lemma 4 applies, we generate the clause (for each $j \in [w]$)

$$\bigvee_{\{u,v\} \notin E(\hat{G})} e_{u,v} \vee \bigvee_{v \in V(G) \setminus A_i} l_{v,j} \vee \bigvee_{v \in A_i^I} \neg l_{v,j}.$$

Finding Gap Graphs

In order to find all gap graphs for $k = 3$ with some number of vertices, we modified the encoding from Section 4 to make use of SMS. We implemented a propagator which checks for each fully defined graph whether it is choosable by the Nullstellensatz. If not, and there are fully defined colorability classes, we run a propagator for Lemma 4. Finding all 106 graphs with 9 vertices that are 3-colorable but not width-7 3-choosable (and thus not 3-choosable by Lemma 3) took about one day. The two smallest are depicted in Figures 2 and 3. The graph in Figure 2 is absolutely smallest in

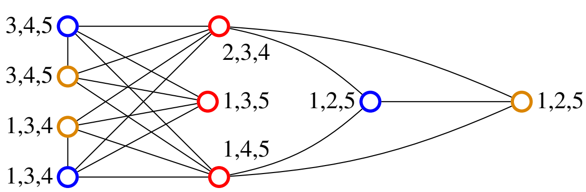


Figure 2: The smallest gap graph for $k = 3$. It has 9 vertices, 19 edges, min degree 3, max degree 6, chromatic number 3, list chromatic number 4, and is Hamiltonian. It is minimal with respect to the number of vertices, as well as the number of edges among graphs with 9 vertices. Drawn 3-colored and with a bad list assignment.

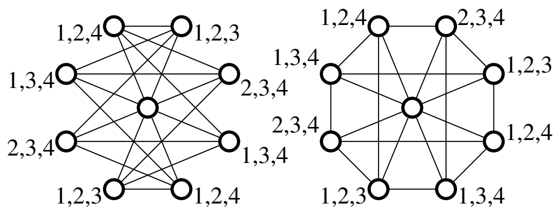


Figure 3: Another gap graph for $k = 3$: 9 vertices, 20 edges, min degree 4, max degree 8, Hamiltonian. It is uniquely minimal with respect to the width of a bad list assignment: only 4 colors are needed. Drawn in two ways: a 3-coloring is obvious on the left; the right has fewer crossings. The list for the central vertex may contain all 4 colors.

terms of number of edges, and it is the only smallest such graph, but requires a bad list assignment of width 5. The graph in Figure 3 has one more edge, but has a bad list assignment of width only 4, which is smallest possible (for 3 colors every list has to be the same). We minimized the number of edges by adding and progressively tightening a cardinality constraint on the number of edges; we similarly tweak the width parameter.

Tables 1 and 2 show a comparison of the various solving approaches and the impact of the width bound on solving time. The tests were conducted on a cluster equipped with Intel Xeon E5-2640 v4 2.40GHz processors running Linux, timeout set to 5 hours. The main takeaway is that the vetting method is the most important optimization. The running time appears exponential in the width parameter.

6 Smallest Planar 4-NS-Oblivious Graphs

Nelsen asked for the smallest non-4-choosable and the smallest 4-NS-oblivious planar graph (Nelsen 2019, Questions 3.21 and 3.22). We were unable to improve Mirzakhani's 63-vertex graph for non-4-choosability but did find a 4-NS-oblivious planar graph with 27 vertices. In the first part of this section, we explain this graph intuitively. In the second part, we describe how we verified the graph's 4-choosability: we used the QBF encoding from Section 4 with non-trivial performance enhancements.

Let G be a planar graph. We call a triangle $\{u, v, w\} \subseteq V$ an *inscribed triangle* if there is a triangle $\{x, y, z\} \subseteq V(G)$ so that $\{(u, x), (u, z), (v, x), (v, y), (w, y), (w, z)\} \subseteq E(G)$ and $\{u, v, w\}$ is a component in $G - \{x, y, z\}$ (see Figure 4). Note that u, v, w all have degree 4 by definition.

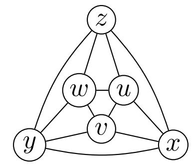


Figure 4: An inscribed triangle. There must be no other edges incident to u, v, and w.

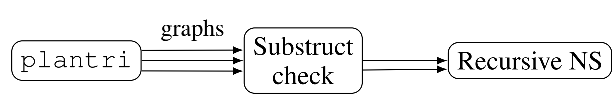
The graph polynomial p_G from Theorem 1 is a homogeneous polynomial: every monomial has degree $|E(G)|$. In a planar triangulation, $|E(G)| = 3n - 6$. Let $\{u, v, w\}$ be a triangle inscribed in $\{x, y, z\}$. The polynomial $p_{u,v,w}$ of the subgraph G_{uvw} induced by $\{u, v, w, x, y, z\}$ is up to sign:

$$p_{u,v,w} = (u-v)(v-w)(w-u)(x-y)(y-z)(z-x)$$
$$(x-u)(x-v)(y-v)(y-w)(z-w)(z-u)$$

It is not too hard to see that $p_{u,v,w}$ does not contain any monomial where u, v, w all have degree 3. Therefore, if one takes a planar triangulation and inserts at least 7 inscribed triangles, any monomial will contain at least 7 variables of degree at most 2. However, the degree of any monomial is $3n - 6$, and so there must be some variable of degree at least 4, and Theorem 1 yields no result for 3-choosability. The smallest planar graph that has 7 or more faces has 6 vertices[4]; with the 7 inscribed triangles we get 27 vertices. There are two triangulations with 6 vertices up to isomorphism, and the argument works for both of them, giving rise to two different NS-oblivious graphs. One is drawn in Figure 6. Both of them are 4-choosable; in the sequel we prove this for the graph from Figure 6. We also prove that no smaller 4-NS-oblivious planar graphs exist.

4-NS-Oblivious Planar Graphs Have ≥ 27 Vertices

We have extended Nelsen's method to rule out the existence of 4-NS-oblivious graphs with 25 or 26 vertices. Schematically, our method looks as follows.



We use `plantri` to generate all triangulations with minimum degree 4 (a necessary condition for a vertex-minimal 4-NS-oblivious graph), filter out those that contain a certain *substructure* (an embedded "jewel" subgraph; its absence is also a necessary condition), and only test the rest with full-blown Nullstellensatz. Our expansion of the graph polynomial for NS is organized recursively to improve performance; all in order to minimize time spent expanding polynomials, the bottleneck of Nelsen's method. Both improvements use an elementary lemma about polynomials.

[4]By Euler's formula #faces $\leq 2n - 4$.

	no opt			nullstellensatz			vetting			vet + ns		
	3	4	5	3	4	5	3	4	5	3	4	5
6	0.05	0.18	0.35	0.03	0.08	0.21	0.01	0.02	0.02	0.01	0.02	0.02
7	0.17	1.27	4.02	0.11	0.73	2.83	0.05	0.09	0.10	0.04	0.08	0.11
8	1.04	16.28	112.41	0.64	10.12	85.56	0.21	0.56	0.88	0.20	0.54	0.94
9	4.33	196.32	2966.41	3.55	153.16	2494.83	0.92	3.79	7.02	0.81	3.63	6.62
10	33.18	3536.28	-	29.44	3013.03	-	5.34	56.88	154.10	4.87	55.43	149.85
11	302.26	-	-	289.94	-	-	51.85	1289.97	4000.80	50.74	1224.10	3997.48

Table 1: Results for bipartite graphs (seconds, - means timeout). Rows give the number of vertices, columns give the width of a bad list assignment and whether the propagators for the Nullstellensatz and vetting method are used.

	no opt			nullstellensatz			vetting			vet + ns		
	4	5	6	4	5	6	4	5	6	4	5	6
7	0.25	1.52	4.45	0.13	0.93	3.35	0.06	0.06	0.24	0.04	0.05	0.11
8	2.61	56.99	520.47	1.27	17.59	247.26	0.36	1.03	4.95	0.26	0.83	4.88
9	34.39	3705.66	-	28.76	2503.89	-	6.78	432.22	11606.48	4.68	394.97	9858.11

Table 2: Results for tripartite graphs (seconds, - means timeout). Rows give the number of vertices, columns give the width of a bad list assignment and whether the propagators for the Nullstellensatz and vetting method are used.

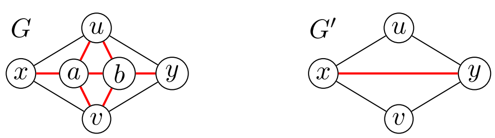


Figure 5: The jewel and diamond graphs from Lemma 6.

Lemma 5. *Let $p(\vec{x},\vec{y}) = \sum_i m_i(\vec{x})r_i(\vec{y})$, where the m_i are monomials over $\vec{x}$, $m_i \neq m_j$, $i \neq j$, the r_i are polynomials, and $\vec{x} \cap \vec{y} = \emptyset$. If some r_i contains a monomial M, then p contains the monomial $m_i M$ with the same coefficient.*

Lemma 6. *Let G be a smallest NS-oblivious planar triangulation. Then, G does not contain a* jewel*: a 4-cycle x, u, y, v separating two degree-4 vertices a, b without the edges xy, ay, bx (see Figure 5).*

Proof. Let G' be G with the jewel replaced with a diamond, i.e., $G' = G - \{a, b\} + \{xy\}$ (see Figure 5 right). Then $p_G = \sum_{0\le i,j\le 3}^{i+j\le 5} a^i b^j r_{ij} + a^3 b^3 p_{G'}$ for some r_{ij} and by Lemma 5 an NS-suitable monomial transfers from $p_{G'}$ to p_G. □

To find jewels, we search for pairs of adjacent vertices of degree 4 and check whether their neighborhood forms a jewel. This is much faster than checking NS: Table 4 shows that the time spent for jewel search is negligible. Table 3 shows that only 3%-5% of the graphs have no jewel and need an NS check; considering that the jewel check is negligible, this amounts to a 20-30x speedup.

The other improvement is the recursive evaluation of the graph polynomial, also based on Lemma 5. Suppose that edges are expanded one by one, and all edges incident to some vertex v have been expanded. Then $p_G = \sum_{i=0}^{3} v^i p_i$ for some polynomials p_i that do not contain v, and we can expand the p_i separately by Lemma 5. We branch this way

n	#graphs	#no jewel	%no jewel
22	2194439398	104294439	4.8%
23	12941995397	561008199	4.3%
24	76890024027	3036531555	4.0%
25	459873914230	16528499985	3.6%
26	2767364341936	90438201894	3.3%

Table 3: Number of triangulations: all with min degree 4, the number, and the proportion of those with no jewel.

n	J total	J ratio	NS total	NS ratio
22	0.4	6.1%	6.8	93.9%
23	2.7	3.5%	74.1	96.5%
24	7.8	2.6%	289.5	97.4%
25	74.9	1.7%	4237.1	98.3%

Table 4: Total and relative time (hours) spent searching for jewels and checking the Nullstellensatz.

whenever a vertex is fully expanded; for most planar graphs we find a suitable monomial with few branching steps, saving space and time by reducing the size of the intermediate polynomial. For example for all triangulations with $n = 17$ vertices, the branching implementation takes only 2 seconds while direct expansion takes 6.5 minutes. We conclude:

Theorem 2. *Smallest planar 4-NS-oblivious graphs have 27 vertices.*

Proof of 4-Choosability of the Graph in Figure 6

Due to its size, solving the direct QBF encoding $F^w_{n,4}$ from Section 4 instantiated with $n = 27$, the edges of the graph from Figure 6, and the best possible width bound $w = 13$ from Lemma 3 is not feasible. The vetting method from Sec-

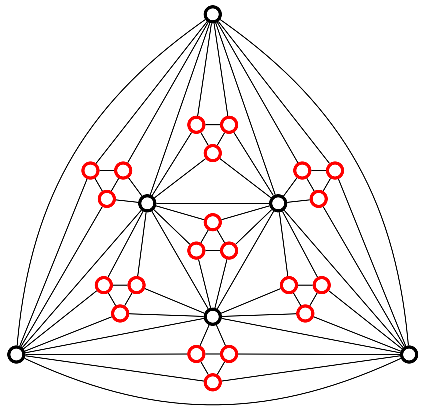


Figure 6: A smallest 4-NS-oblivious planar graph: 27 vertices, 75 edges, min degree 4, max degree 12, chromatic number 3, list chromatic number 4, not Hamiltonian. It arises by nesting inscribed triangles (red).

tion 5 fails as well: there are too many colorability classes that pass the test. The central result that allows us to solve the QBF encoding is Lemma 9, which shows how to reduce the question of choosability of a large graph G with inscribed triangles to a restricted version of choosability of a much smaller graph G'.

Lemma 7. *Let L be a list assignment to the vertices of the triangle K_3 in which every list has size at least 2. K_3 is not L-colorable if and only if all lists have size 2 and are equal.*

Proof. K_3 is not 2-colorable. If L's lists are not all equal, L has width ≥ 3 and Hall's theorem gives an L-coloring. □

Lemma 8. *Let G be a graph and $\{u, v, w\}$ be an inscribed triangle in G and $L_{|\{u,v,w\}}$ be a 4-full list assignment to the vertices u, v, w. At most one coloring $C_{|x,y,z}$ of the neighboring vertices $\{x, y, z\}$ cannot be extended to u, v, w.*

Proof. Let C be a coloring of $\{x, y, z\}$. Whether C can be extended to $\{u, v, w\}$ is the same question as whether $\{u, v, w\}$ is L'-colorable for $L'(u) = L(u) \setminus \{ C(z), C(x) \}$, $L'(v) = L(v) \setminus \{ C(x), C(y) \}$, and $L'(w) = L(w) \setminus \{ C(y), C(z) \}$. By Lemma 7, $\{u, v, w\}$ is not L'-colorable if, and only if, $L'(u) = L'(v) = L'(w) = \{a, b\}$. Therefore $L(u) = \{a, b, C(x), C(z)\}$, $L(v) = \{a, b, C(x), C(y)\}$, $L(w) = \{a, b, C(y), C(z)\}$. Then $C(x), C(y), C(z)$ must all be different, and are uniquely defined by $\{C(x)\} = L(u) \cap L(v) \setminus L(w)$, $\{C(y)\} = L(v) \cap L(w) \setminus L(u)$, $\{C(z)\} = L(w) \cap L(u) \setminus L(v)$. □

This leads to a reduction in the number of vertices by way of replacing inscribed triangles with forbidden partial colorings. We formalize this process by the notion of "patterns."

Let W be the set of colors and x, y, z an outer triangle. The set of *patterns* over $\{x, y, z\}$ is given by $\mathcal{P}(x, y, z) = \{ c : \{x, y, z\} \to W \mid c(x) \neq c(y) \neq c(z) \neq c(x) \}$, i.e., the set of proper colorings of the triangle. A coloring C of the whole graph G *follows* a pattern $c \in \mathcal{P}(x, y, z)$ if $C|_{\{x,y,z\}} = c$; otherwise C *avoids* c.

According to Lemma 8, an inscribed triangle amounts to one forbidden pattern. Choosability then boils down to checking all possible patterns, as formalized in Lemma 9.

Lemma 9. *Let G be a graph, $\mathcal{I} = \{I_1, \ldots, I_d\}$ a set of inscribed triangles in G, and $\mathcal{O} = \{O_1, \ldots, O_d\}$ the set of the corresponding outer triangles, such that the inscribed triangles are pairwise disjoint and for each pair $I \in \mathcal{I}, O \in \mathcal{O}$ we have $I \cap O = \emptyset$. If for all 4-full list assignments L' of $G' = G - \bigcup_{I \in \mathcal{I}}$ and all tuples of patterns $(c_1, \ldots, c_d) \in \prod_{O \in \mathcal{O}} \mathcal{P}(O)$ there is an L'-coloring C' that avoids all of the patterns $c_1, \ldots, c_d$, then G is 4-choosable.*

Proof. Let L be any list assignment of G. By Lemma 8, for any inscribed triangle I_i at most one pattern on its outer triangle O_i is not extensible to I_i. Because the triangles are disjoint, we can apply Lemma 8 to each triangle separately, and for any given list assignment L, we can collect a tuple of forbidden patterns $c_1, \ldots, c_d$ on the respective outer triangles. By assumption, G' is $L|_{V(G')}$-colorable with a coloring C' that avoids all of these patterns simultaneously, so C' can be extended to all I_i to obtain an L-coloring of G. □

From Observation 1 we immediately obtain that colorability classes overlap somewhere in the big graph G. By a refined version of the same argument, we can force colorability classes to overlap on vertices of the reduced graph G'.

Lemma 10. *Let G and G' be as defined in Lemma 9. If there is a list assignment L' and patterns $c_1, \ldots, c_d$ s.t. each proper L'-coloring of G' follows at least one pattern c_i, then there is a bad list assignment $\hat{L}$ of G' each pair of whose colorability classes A_1, A_2 intersect in some vertex of G'.*

Proof. Assume, there exists a bad list assignment L' and patterns $c_1, \ldots, c_d$ where there are colorability classes $A'_i \cap A'_j = \emptyset$. We claim that unifying A'_i and A'_j (renaming color j to i) yields a bad list assignment $\hat{C}$. Towards contradiction, consider a proper $\hat{L}$-coloring $\hat{C}$ that avoids all patterns, and let C' be the implied L'-coloring (where some vertices are recolored from i back to j). Since $\hat{C}$ is proper, C' is also proper. Suppose C' follows a pattern on some $\{x, y, z\}$ that $\hat{C}$ avoids: that can only happen if $C'(x) = i$ and $C'(y) = j$. But then, $\hat{C}(x) = \hat{C}(y)$, a contradiction. □

Similarly to Corollary 1, we get a tighter bound on width by counting intersection witnesses in G'.

Corollary 2. *Let G' be as above, with $|V(G')| = 6$. If G' has a bad list assignment L' taking into account patterns on inscribed triangles, then L' has width ≤ 8. If L' has width 8, then $|A_i| = 3$ for each colorability class A_i.*

Proof. Suppose, $p = 8$. By Lemma 10, each pair of the 8 colorability classes intersects on one of the 6 vertices of G'. Like in Lemma 3, that means $\forall i \; |A_i| \geq \lceil \frac{8-1}{4-1} \rceil = 3$. Now, $\sum_{A_i \in A} |A_i| = 6 \cdot 4 = 24$, which means that $|A_i| = 3$. An analogous argument rules out $p > 8$. □

Encoding Pattern-Avoiding Choosability

Given a graph G together with a set of inscribed triangles $\mathcal{I}$ and the outer triangles $\mathcal{O}$, we construct a QBF encoding pattern-avoiding choosability. Some constraints are reused from Section 4. Consider the formula

$$\forall L, P \exists C \; well(L, P, f) \rightarrow (col(L, C) \wedge avoids(P, C)),$$

where $well(L, P, f) = full(L, f) \wedge ptrn(P) \wedge overlaps(L)$,

$$ptrn(P) = \bigwedge_{O \in \mathcal{O}}^{v \in O} \left(\sum_{j \in [w]} p^j_{O,v} = 1 \wedge \bigwedge_{j \in [w]}^{\substack{u \in O \\ u < v}} \neg p^j_{O,u} \vee \neg p^j_{O,v} \right)$$

$$avoids(P, C) = \bigwedge_{O \in \mathcal{O}} \bigvee_{v \in O} \bigwedge_{j \in [w]} \left(\neg c^j_v \vee \neg p^j_{O,v} \right),$$

The set of variables $P = \{ p^j_{O,v} \mid O \in \mathcal{O}, v \in O, j \in [w] \}$ encodes that v is assigned color j in the pattern of the outer triangle O.

To solve the resulting formula efficiently, we adapt to pattern-avoiding choosability and run as preprocessor the vetting method from Section 5 (203 seconds). The formula is then solved in 3 seconds by CQesto (Janota 2018).

7 Conclusion

This paper highlights the importance of hybrid SAT and QBF-based methods with custom propagators as opposed to just straightforward encodings.

With our approach we enumerated all gap graphs on 9 vertices (3-colorable but not 3-choosable). Furthermore, we found a smallest 27-vertex planar graph that is 4-NS-oblivious and were able to verify its 4-choosability using an advanced encoding. Finding a smallest non-4-choosable planar graph remains an open challenge; our discovery of inscribed triangles and the jewel structure (Figure 5) could prove helpful in this search as well.

An intriguing avenue for future research is to design propagators based on mathematical principles like the Nullstellensatz for general QBF solving.

Acknowledgments

The authors acknowledge support by the Austrian Science Fund (FWF), via the projects 10.55776/COE12 and 10.55776/P36688.

References

Alon, N.; and Tarsi, M. 1992. Colorings and orientations of graphs. *Comb.*, 12(2): 125–134.

Appel, K.; Haken, W.; and Koch, J. 1977. Every planar map is four colorable. Part II: Reducibility. *Illinois Journal of Mathematics*, 21(3): 491–567.

Biere, A.; Faller, T.; Fazekas, K.; Fleury, M.; Froleyks, N.; and Pollitt, F. 2024. CaDiCaL 2.0. In Gurfinkel, A.; and Ganesh, V., eds., *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part I*, volume 14681 of *Lecture Notes in Computer Science*, 133–152. Springer.

Brinkmann, G.; and McKay, B. 2007. *Fast generation of planar graphs*. University of Kragujevac.

Dvořák, Z. 2024. A Strengthening and an Efficient Implementation of Alon-Tarsi List Coloring Method. *The Electronic Journal of Combinatorics*, 31.

Enomoto, H.; Ohba, K.; Ota, K.; and Sakamoto, J. 2002. Choice number of some complete multi-partite graphs. *Discrete Mathematics*, 244(1): 55–66.

Erdős, P.; Rubin, A. L.; and Taylor, H. 1979. Choosability in graphs. *Congr. Numer*, 26(4): 125–157.

Fazekas, K.; Niemetz, A.; Preiner, M.; Kirchweger, M.; Szeider, S.; and Biere, A. 2024. Satisfiability Modulo User Propagators. *J. Artif. Intell. Res.*, 81: 989–1017.

Fichte, J. K.; Hecher, M.; Berre, D. L.; and Szeider, S. 2023. The silent (R)evolution of SAT. *Communications of the ACM*, 66(6): 64–72.

Janota, M. 2018. Circuit-Based Search Space Pruning in QBF. In Beyersdorff, O.; and Wintersteiger, C. M., eds., *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings*, volume 10929 of *Lecture Notes in Computer Science*, 187–198. Springer.

Janota, M.; Kirchweger, M.; Peitl, T.; and Szeider, S. 2025. Breaking Symmetries in Quantified Graph Search: A Comparative Study. In *AAAI 2025: The 39th Annual AAAI Conference on Artificial Intelligence*.

Jordan, C.; Klieber, W.; and Seidl, M. 2016. Non-CNF QBF Solving with QCIR. In Darwiche, A., ed., *Beyond NP, Papers from the 2016 AAAI Workshop.*, volume WS-16-05 of *AAAI Workshops*. AAAI Press.

Kierstead, H. A. 2000. On the choosability of complete multipartite graphs with part size three. *Discrete Mathematics*, 211(1-3): 255–259.

Kirchweger, M.; and Szeider, S. 2024. SAT Modulo Symmetries for Graph Generation and Enumeration. *ACM Trans. Comput. Log.*, 25(3): 1–30.

Marques-Silva, J.; Lynce, I.; and Malik, S. 2021. Conflict-Driven Clause Learning SAT Solvers. In Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds., *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, 133–182. IOS Press.

Mirzakhani, M. 1996. A small non-4-choosable planar graph. *Bull. Inst. Combin. Appl*, 17(15-18): 3.

Nelsen, L. L. 2019. *Computational methods for graph choosability and applications to list coloring problems*. Ph.D. thesis, PhD thesis, University of Colorado Denver.

Noel, J. A.; Reed, B. A.; and Wu, H. 2012. A Proof of a Conjecture of Ohba. *Journal of Graph Theory*, 79.

Robertson, N.; Sanders, D.; Seymour, P.; and Thomas, R. 1997. The Four-Colour Theorem. *Journal of Combinatorial Theory, Series B*, 70(1): 2–44.

Sinz, C. 2005. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In van Beek, P., ed., *Principles and Practice of Constraint Programming - CP 2005,* *11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings*, volume 3709 of *Lecture Notes in Computer Science*, 827–831. Springer Verlag.

Thomassen, C. 1994. Every planar graph is 5-choosable. *Journal of Combinatorial Theory Series B*, 62(1): 180–181.

Voigt, M. 1993. List colourings of planar graphs. *Discrete Mathematics*, 120(1-3): 215–219.

West, D. B. 2020. *Combinatorial Mathematics*. Cambridge University Press.