# Graph Coloring Is Hard on Average for Polynomial Calculus

Jonas Conneryd
Department of Computer Science, Lund University

LUND UNIVERSITY

## Introduction

Let $\mathbb{G}(n, d/n)$ be the distribution over $n$-vertex graphs where each edge appears independently with probability $d/n$. It is well-known that the chromatic number of a graph $G = (V, E) \sim \mathbb{G}(n, d/n)$ is tightly concentrated around $d/\log d$ w.h.p. as $n \to \infty$. But, given an actual sample $G \sim \mathbb{G}(n, d/n)$, how hard is it for state-of-the art algorithmic paradigms to determine its chromatic number, even approximately?

In the paper [CdRN+23] we show that w.h.p. a wide range of algebraic graph coloring algorithms take exponential time to rule out that a graph $G \sim \mathbb{G}(n, d/n)$ is not 3-colorable, even though its actual chromatic number is $\sim d/\log d \gg 3$. Our results provide an abundance of hard instances of graph coloring for algebraic algorithms, even though these algorithms perform remarkably well on practical benchmarks. Upcoming work [CRS25] extends these hardness results to an encoding of graph coloring to which the results of [CdRN+23] do not apply.

## Background

### Graph Coloring and Polynomials

We are interested in *algebraic* algorithms, which solve the graph coloring problem by encoding it as a system of polynomials whose common roots correspond to $k$-colorings of the underlying graph. There are two natural such encodings: the *Boolean encoding* and the *roots-of-unity encoding*.

**Boolean encoding:** $x_{v,i} = 1 \iff$ *vertex $v$ gets color $i$*

$$
\begin{array}{lll}
\sum_{i=1}^{k} x_{v,i} - 1, & \forall v \in V & \textit{every vertex gets a color} \\
x_{v,i} \cdot x_{v,i'}, & \forall v \in V,\ i \neq i' & \textit{no vertex gets} > 1 \textit{ color} \\
x_{u,i} \cdot x_{v,i}, & \forall (u,v) \in E & \textit{no monochromatic edges} \\
x_{v,i}^2 - x_{v,i}, & \forall v, i & \textit{Boolean axioms}
\end{array}
$$

**Roots-of-unity encoding:** $y_v = \omega^i \iff$ *vertex $v$ gets color $i$*

$$
\begin{array}{lll}
y_v^k - 1 & v \in V & \textit{every vertex gets a color} \\
\sum_{j=0}^{k-1} y_u^j y_v^{k-1-j} & (u,v) \in E & \textit{no monochromatic edges}
\end{array}
$$

### Proof Complexity

Our result is a lower bound on the running time on a family of algorithms for the graph coloring problem. We prove it by interpreting the algorithm as providing a *proof* of its output. The method of reasoning the algorithm uses can thus be seen as a collection of inference rules in a proof system $P$. A lower bound on the size of any $P$-proof that a graph $G \sim \mathbb{G}(n, d/n)$ is not 3-colorable then implies a lower bound on the running time of the algorithm.

### Polynomial Calculus

The reasoning used by the algorithms we are interested in is captured by the *polynomial calculus* proof system. Polynomial calculus proves a that set of polynomials $\Gamma = \{p_1, \ldots, p_m\}$ has no common root by deriving new polynomials in the ideal $\langle \Gamma \rangle$ through two derivation rules:

$$
\text{Linear combination: } \frac{p \quad q}{\alpha p + \beta q} \quad \alpha, \beta \in \mathbb{F}
$$

$$
\text{Multiplication: } \frac{p}{x \cdot p} \quad x \text{ any variable}
$$

A *polynomial calculus refutation* of $\Gamma$ is a derivation of the constant polynomial 1. There are two main complexity measures in polynomial calculus: *size*, which is the total number of monomials in the proof lines (with multiplicities); and *degree*, which is the largest degree among the monomials in the proof lines.

## Results

Our main result is a strongly exponential average-case polynomial calculus size lower bound for refuting that a graph $G$ sampled from $\mathbb{G}(n, d/n)$ is 3-colorable.

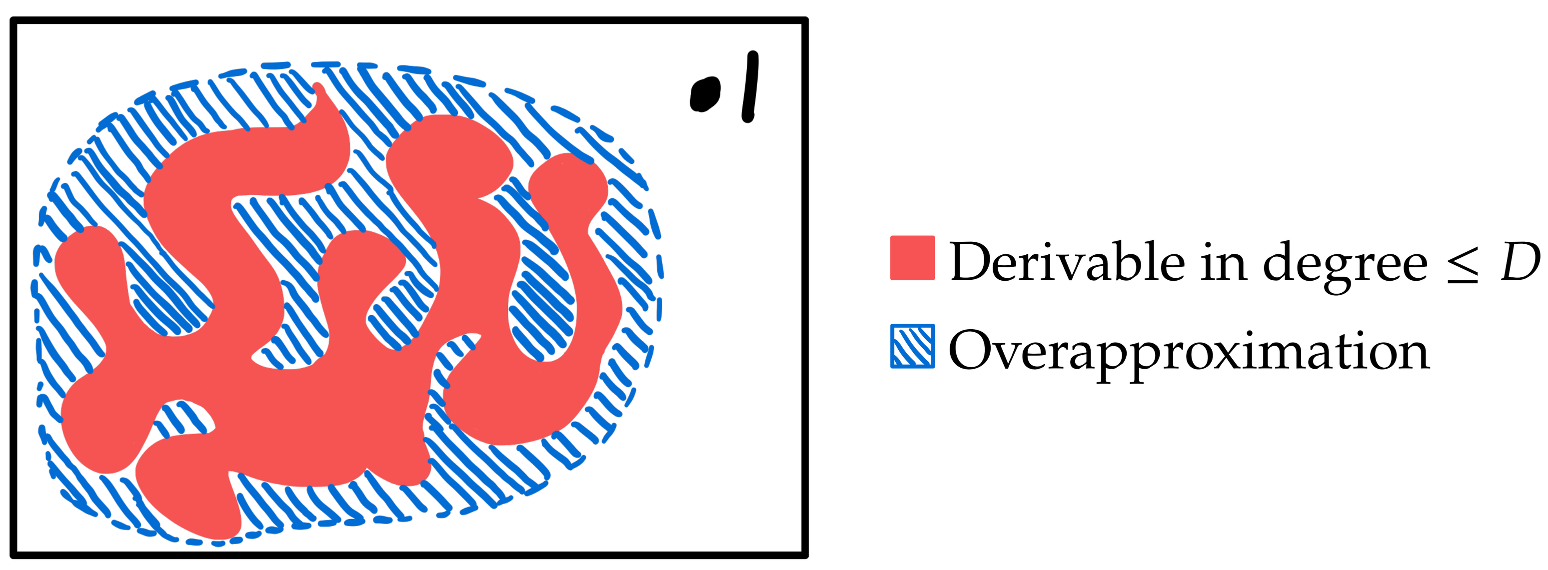
**Theorem.** *If $G \sim \mathbb{G}(n, d/n)$ and $d$ is constant, then with probability $1 - o(1)$ polynomial calculus requires size $\exp(\Omega(n))$ to refute that $G$ is 3-colorable when the problem is expressed in either the Boolean encoding or the roots-of-unity encoding.*

### Proof Techniques

For the Boolean encoding it suffices to prove a $\Omega(n)$ degree lower bound, which implies an $\exp(\Omega(n))$ size lower bound by known results. The lower bound for the roots-of-unity encoding is also based on a degree lower bound but requires additional work.
To prove a polynomial calculus degree lower bound, one defines a so-called *pseudo-reduction operator* $R$ on polynomials such that

- $R(p) = 0$, for each input polynomial $p \in \Gamma$;
- $R(p) + R(q) = R(p + q)$;
- if $R(p) = 0$ then $R(x \cdot p) = 0$, for all $p$ of degree $\leq D - 1$;
- $R(1) = 1$.

The idea is to overapproximate the set of polynomials derivable in degree at most $D$ by the kernel of $R$, and in particular to show that 1 is not in the kernel of $R$.




## References

[CdRN+23] J. Conneryd, S. F. de Rezende, J. Nordström, S. Pang, and K. Risse, Graph colouring is hard on average for polynomial calculus and Nullstellensatz, in *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS '23)*, Nov. 2023, pp. 1–11.

[CRS25] J. Conneryd, K. Risse, and D. Sokolov, Graph coloring is hard for polynomial calculus over roots of unity, 2025.