

# Trimming Graphs Using Clausal Proof Optimization

**Marijn J.H. Heule**

**Carnegie  
Mellon  
University**

Constraint Programming

October 1, 2019

Introduction

Clausal Proof Optimization

Observed Patterns in  $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

Small UD Graphs with Chromatic Number 5

Conclusions and Future Work

# Introduction

Clausal Proof Optimization

Observed Patterns in  $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

Small UD Graphs with Chromatic Number 5

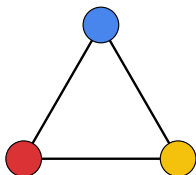
Conclusions and Future Work

# Chromatic Number of the Plane

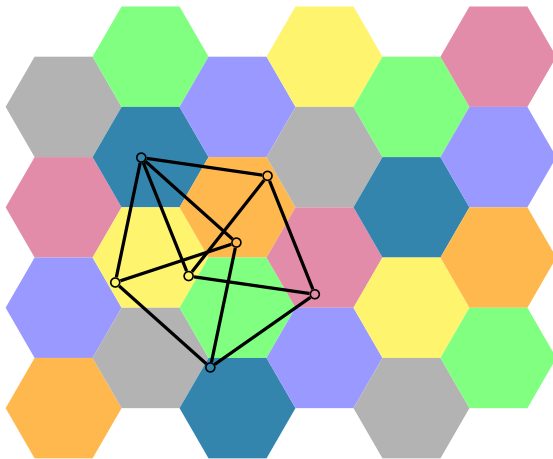
The Hadwiger-Nelson problem:

How many colors are required to color the plane such that each pair of points that are exactly 1 apart are colored differently?

The answer must be three or more because three points can be mutually 1 apart—and thus must be colored differently.



## Bounds since the 1950s

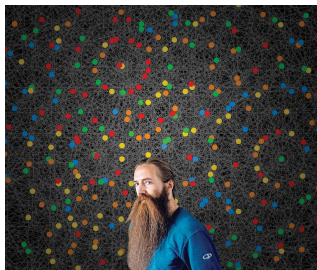


- The Moser Spindle graph shows the lower bound of 4
- A coloring of the plane showing the upper bound of 7

# First progress in decades

Recently enormous progress:

- Lower bound of 5 [DeGrey '18] based on a 1581-vertex graph
- This breakthrough started a polymath project
- Improved bounds of the fractional chromatic number of the plane



# First progress in decades

Recently enormous progress:

- Lower bound of 5 [DeGrey '18] based on a 1581-vertex graph
- This breakthrough started a polymath project
- Improved bounds of the fractional chromatic number of the plane



Quanta magazine

Physics Mathematics

業餘數學家為一道填色難題帶來突破！

2018/4/26 · TNL · 四色定理 · 填色問題 · 數學

Раскраска для математиков

Как покрасить плоскость?

WIRED

Marijn Heule, a computer scientist at the University of Texas, Austin, found one with just 874 vertices. Yesterday he lowered this number to 826 vertices.

We found smaller graphs with SAT:

- 874 vertices on April 14, 2018
- 803 vertices on April 30, 2018
- 610 vertices on May 14, 2018

## Validation

**Check 1:** Are two given points exactly 1 apart? For example:

- $\left( \frac{19+3\sqrt{5}}{16}, \frac{5\sqrt{15}-7\sqrt{3}}{16} \right)$
- $\left( \frac{135+21\sqrt{5}-7\sqrt{33}+3\sqrt{165}}{96}, \frac{33\sqrt{15}-49\sqrt{3}-21\sqrt{11}-3\sqrt{55}}{96} \right)$

**Our method:** An approach based on Groebner basis theory developed by Armin Biere, Manuel Kauers, Daniela Ritirc



## Validation

**Check 1:** Are two given points exactly 1 apart? For example:

- $\left( \frac{19+3\sqrt{5}}{16}, \frac{5\sqrt{15}-7\sqrt{3}}{16} \right)$
- $\left( \frac{135+21\sqrt{5}-7\sqrt{33}+3\sqrt{165}}{96}, \frac{33\sqrt{15}-49\sqrt{3}-21\sqrt{11}-3\sqrt{55}}{96} \right)$

**Our method:** An approach based on Groebner basis theory developed by Armin Biere, Manuel Kauers, Daniela Ritirc

**Check 2:** Given a graph  $G$ , has it chromatic number  $k$ ?

**Our method:** Construct two Boolean formulas: one asking whether  $G$  can be colored with  $k - 1$  colors (must be UNSAT) and one asking whether  $G$  can be colored with  $k$  colors (SAT).

# Computer-Aided Mathematics

Fields Medalist Timothy Gowers stated that mathematicians would like to use three kinds of technology [Big Proof 2017]:

- Proof Assistant Technology
  - Prove any lemma that a graduate student can work out
- Proof Search Technology
  - Automatically determine whether a conjecture holds
  - This talk: Find small counter-examples
- Proof Checking Technology
  - Mechanized validation of all details

Introduction

Clausal Proof Optimization

Observed Patterns in  $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

Small UD Graphs with Chromatic Number 5

Conclusions and Future Work

## Extracting Subgraphs from a Proof of Unsatisfiability

The validation method to check whether a graph has (at least) chromatic number  $k$  construct a SAT formula asking whether the graph  $G$  can be colored with  $k - 1$  colors.

The resulting formula is **unsatisfiable**.

Most SAT solvers can emit a **proof of unsatisfiability**.

Proof checkers can extract an **unsatisfiable core** of the problem, which represents a **subgraph** of  $G$ .

# Clausal Proofs of Unsatisfiability

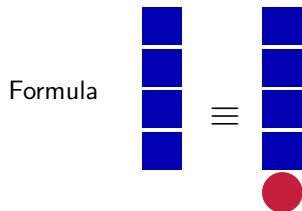
Formula



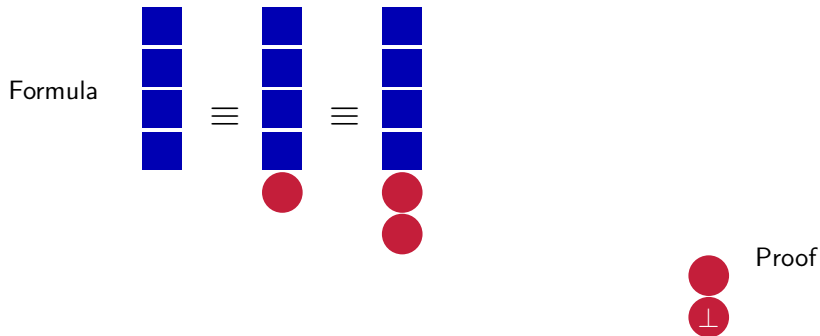
Proof



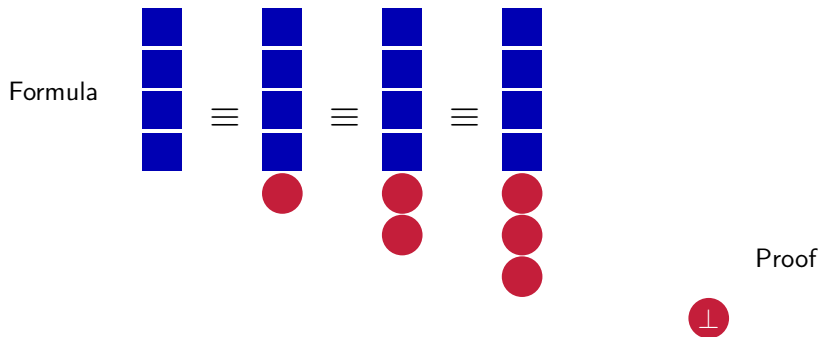
# Clausal Proofs of Unsatisfiability



# Clausal Proofs of Unsatisfiability

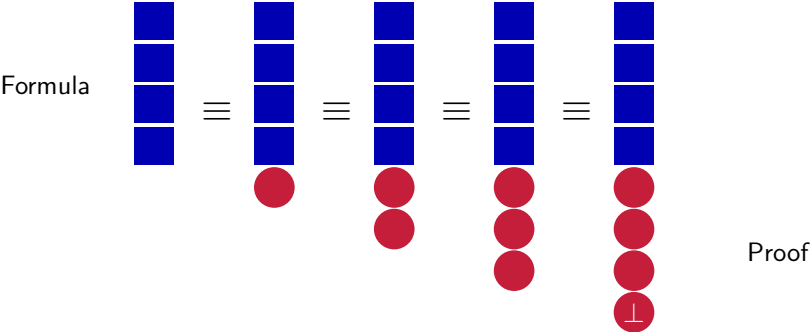


# Clausal Proofs of Unsatisfiability

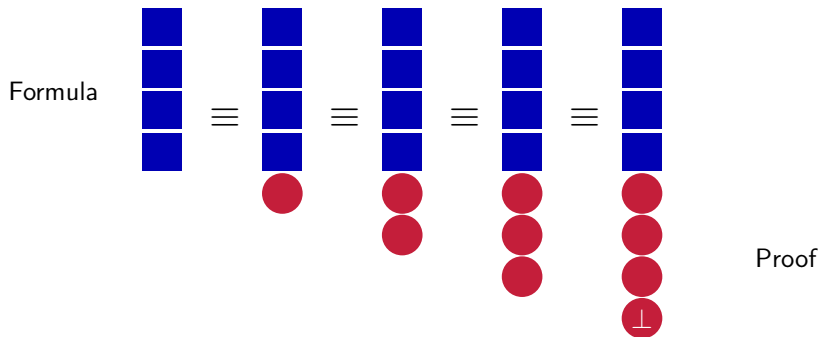




# Clausal Proofs of Unsatisfiability



# Clausal Proofs of Unsatisfiability



- Checking the redundancy of a clause in **polynomial time**
- Clausal proofs are **easy to emit** from modern SAT solvers
- A clausal proof usually covers **many resolution proofs**

# Proof Checking Techniques Advances

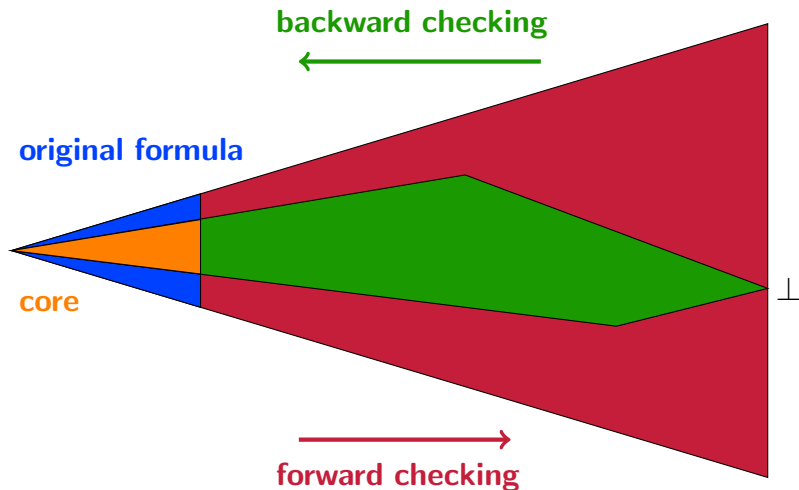
Proof checking techniques have **improved significantly** in recent years.

Clausal proofs of **petabytes** in size can now be validated.

Long-standing open math problems—including the Erdős discrepancy problem, the Boolean Pythagorean triples problem, and Schur number five—have been solved with SAT and their proofs have been constructed and validated.

Efficient validation can even be achieved with a **formally verified** checker.

# Backward Proof Checking: Remove Redundancy



## OptimizeProof

The order of the clauses in the proof and the order of the literals in clauses have a big impact on reduced proof.

- Optimize the proof by checking it **multiple times**;
- Each iteration uses the **reduced proof**; and
- Clauses and literals are **shuffled**.

## OptimizeProof

The order of the clauses in the proof and the order of the literals in clauses have a big impact on reduced proof.

- Optimize the proof by checking it **multiple times**;
- Each iteration uses the **reduced proof**; and
- Clauses and literals are **shuffled**.

Shuffling of clauses is somewhat **limited**:

- A clause must occur **after** all clauses on which it depends;
- A clause must occur **before** all clauses that depend on it.

## OptimizeProof

The order of the clauses in the proof and the order of the literals in clauses have a big impact on reduced proof.

- Optimize the proof by checking it **multiple times**;
- Each iteration uses the **reduced proof**; and
- Clauses and literals are **shuffled**.

Shuffling of clauses is somewhat **limited**:

- A clause must occur **after** all clauses on which it depends;
- A clause must occur **before** all clauses that depend on it.

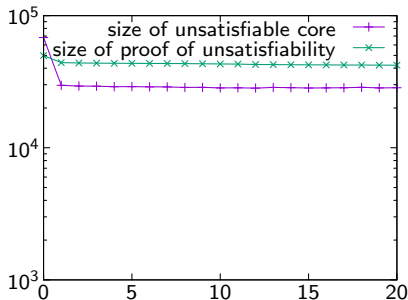
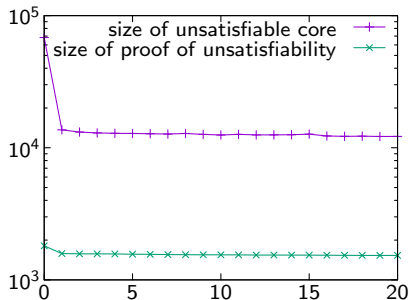
The **OptimizeProof** procedure repeats proof reduction until the size no longer decreases.

## Impact of the Quality of the Proof

The order of the clauses influences the size of the proof

- Solve the problem multiple times with different **clause order**
- Select the **smallest proof** for proof optimization

Left the smallest proof (100 random clause orders) and right the largest proof and 20 iterations of the **OptimizeProof** method



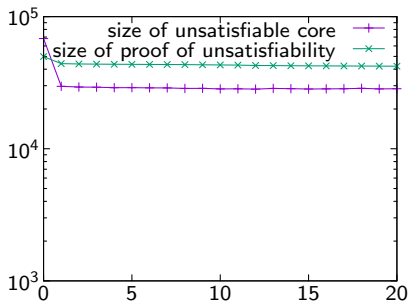
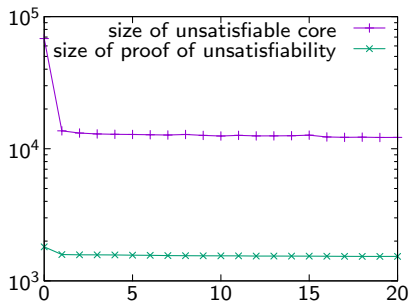


## Impact of the Quality of the Proof

The order of the clauses influences the size of the proof

- Solve the problem multiple times with different **clause order**
- Select the **smallest proof** for proof optimization

Left the smallest proof (100 random clause orders) and right the largest proof and 20 iterations of the **OptimizeProof** method



**the size of the proof correlates with the size of the core**

# TrimFormulaPlain

**Input:** formula  $F$

**Output:** an unsatisfiable core of  $F$

```
1  $F_{\text{core}} := F$ 
2 do
3    $P := \text{Solve} (F_{\text{core}})$ 
4    $P := \text{OptimizeProof} (P, F_{\text{core}})$ 
5    $F_{\text{core}} := \text{ExtractCore} (P, F_{\text{core}})$ 
6 while (progress)
7 return  $F_{\text{core}}$ 
```

**problem:** useful clauses may be removed from  $F_{\text{core}}$

## TrimFormulaInteract

**Input:** formula  $F$

**Output:** an unsatisfiable core of  $F$

```
1  $F_{\text{core}} := F$ 
2 do
3    $P := \text{Solve} (F_{\text{core}})$ 
4    $P := \text{OptimizeProof} (P, F_{\text{core}})$ 
5    $P := \text{OptimizeProof} (P, F)$ 
6    $F_{\text{core}} := \text{ExtractCore} (P, F)$ 
7 while (progress)
8 return  $F_{\text{core}}$ 
```

**solution:** useful clauses can be pulled back in  $F_{\text{core}}$

Introduction

Clausal Proof Optimization

Observed Patterns in  $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

Small UD Graphs with Chromatic Number 5

Conclusions and Future Work

# Graph Operations

Two operations are used to construct bigger and bigger graphs:

- Minkowski sum of  $A$  and  $B$  ( $A \oplus B$ ):  $\{a + b \mid a \in A, b \in B\}$
- Two rotated copies of a graph with a common point

## Example

Let  $A = \{(0, 0), (1, 0)\}$  and  $B = \{(0, 0), (1/2, \sqrt{3}/2)\}$

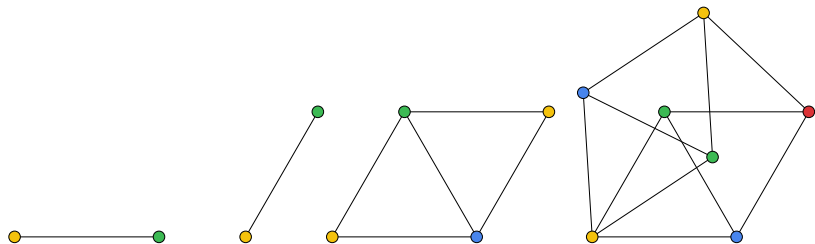
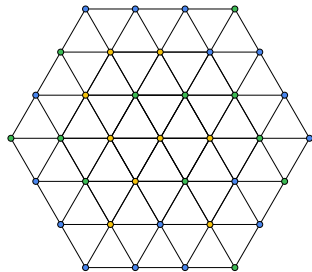


Figure: From left to right: UD-graphs  $A$ ,  $B$ ,  $A \oplus B$ , and the Moser Spindle.

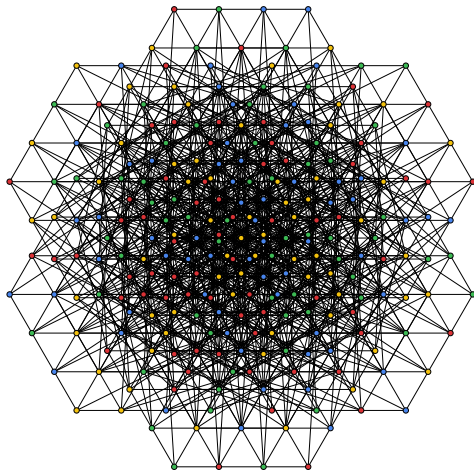
# Small graphs in $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

Graph  $H_i$  is the 6-wheel with all edges of length  $i$ .

Graph  $H'_i$  is a copy of  $H_i$  rotated by 90 degrees.

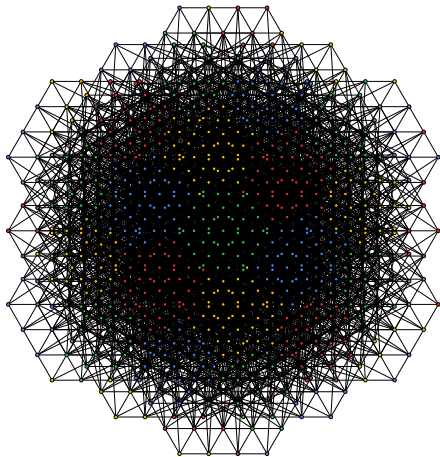


$$H_{\frac{1}{3}} \oplus H_{\frac{1}{3}} \oplus H_{\frac{1}{3}}$$

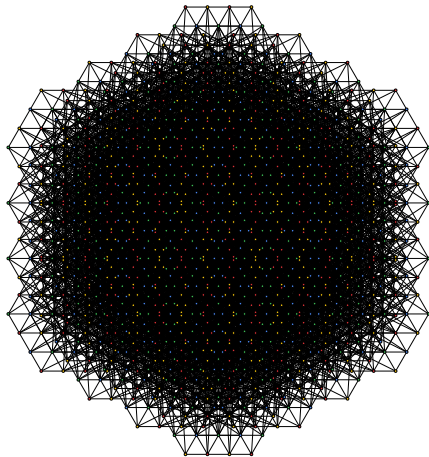


$$H_{\frac{1}{3}} \oplus H_{\frac{1}{3}} \oplus H_{\frac{1}{3}} \oplus H'_{\frac{\sqrt{3}+\sqrt{11}}{6}}$$

# Larger graphs in $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

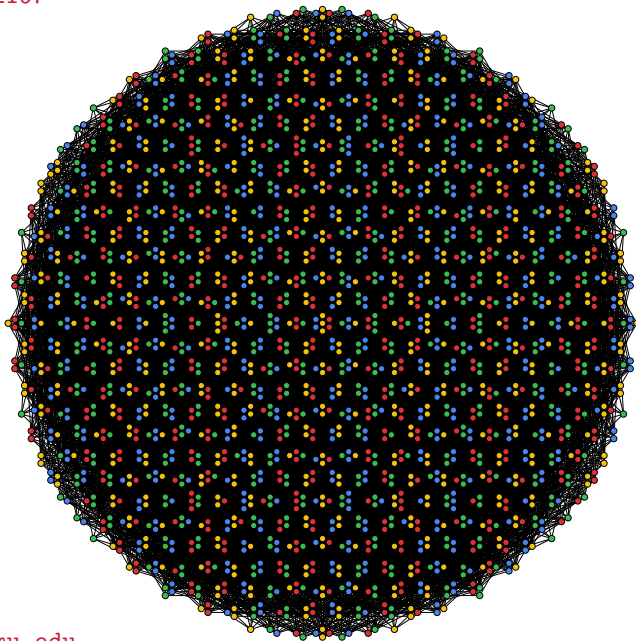


$$\begin{aligned}
 &H_{\frac{1}{3}} \oplus H_{\frac{1}{3}} \oplus H_{\frac{1}{3}} \oplus H'_{\frac{\sqrt{3}+\sqrt{11}}{6}} \\
 &\quad \oplus H'_{\frac{\sqrt{3}+\sqrt{11}}{6}}
 \end{aligned}$$



$$\begin{aligned}
 &H_{\frac{1}{3}} \oplus H_{\frac{1}{3}} \oplus H_{\frac{1}{3}} \oplus H'_{\frac{\sqrt{3}+\sqrt{11}}{6}} \\
 &\quad \oplus H'_{\frac{\sqrt{3}+\sqrt{11}}{6}} \oplus H'_{\frac{\sqrt{3}+\sqrt{11}}{6}}
 \end{aligned}$$

# Graph $G_{2167}$





Introduction

Clausal Proof Optimization

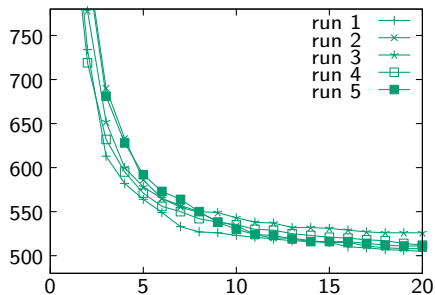
Observed Patterns in  $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

Small UD Graphs with Chromatic Number 5

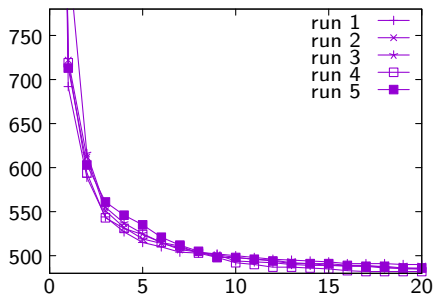
Conclusions and Future Work

# Impact of the Trimming Algorithms

We started with  $G_{2167}$  and reduced it using the proof trimming algorithms: **TrimProofInteract** outperforms **TrimProofPlain**.



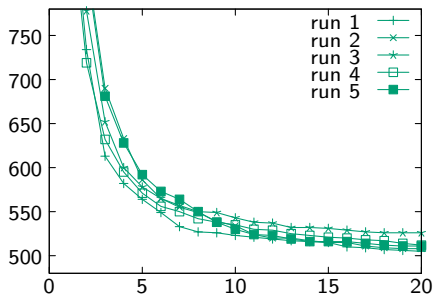
TrimProofPlain



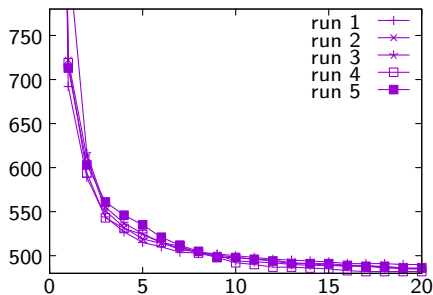
TrimProofInteract

# Impact of the Trimming Algorithms

We started with  $G_{2167}$  and reduced it using the proof trimming algorithms: **TrimProofInteract** outperforms **TrimProofPlain**.



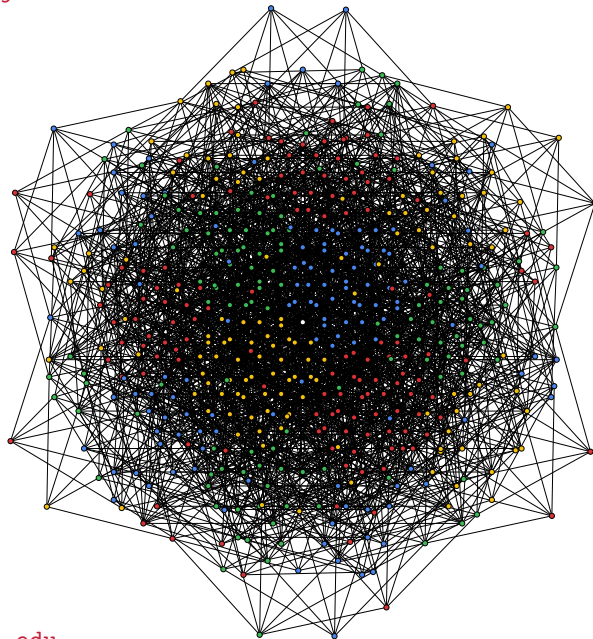
TrimProofPlain



TrimProofInteract

- The smallest graph with the desired properties: 393 vertices
- We added 136 vertices to remove all 4-colorings

# Graph $G_{529}$



Introduction

Clausal Proof Optimization

Observed Patterns in  $\mathbb{Q}[\sqrt{3}, \sqrt{11}] \times \mathbb{Q}[\sqrt{3}, \sqrt{11}]$

Small UD Graphs with Chromatic Number 5

Conclusions and Future Work

## Conclusions and Future Work

Aubrey de Grey showed that the chromatic number of the plane is at least 5 using a **1581-vertex unit-distance graph**.

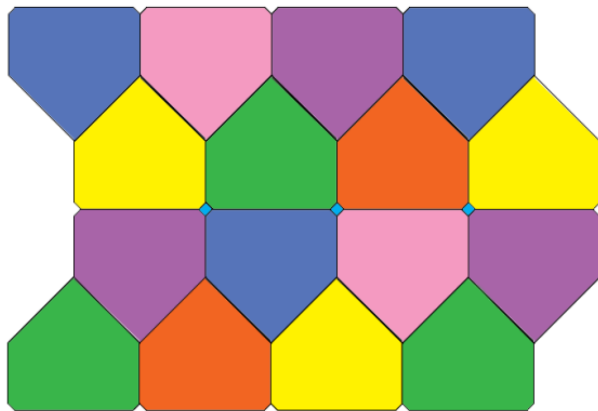
SAT technology can not only **validate** the result, but also **reduce** the size of the graph.

Our proof minimization techniques were able to construct a **529-vertex unit-distance graph** with chromatic number 5.

Open questions regarding unit-distance graphs:

- What is the smallest graph with chromatic number 5?
- Can we compute a graph that is human-understandable?
- Is there such a graph with chromatic number 6 (or even 7)?

## Improve the Upper Bound?



A 7-coloring with one color covering 0.3% of the plane.

[Pritikin 1998]

Can SAT techniques be used to improve the upper bound?

## A Page of God's Book on Theorems

*“For many years now I am convinced that the chromatic number will be 7 or 6. One day, Paul Erdős said that God has an endless book that contains all the theorems and best of their evidence, and to some He shows it for a moment. If I had been awarded such an honor and I would have had a choice, I would have asked to look at the page with the problem of the chromatic number of the plane. And you?”*

Alexander Soifer