Exploring srg(45,22,10,11) and implications for the Ramsey number R(5,5)

J. Braginsky[1] S. Yellenki[1] J. Zhu[1]
with Dr. J. Mackey[2]

[1]Undergraduates
Carnegie Mellon University

[2]Teaching Professor, School of Computer Science
Carnegie Mellon University

57th Southeastern International Conference on Combinatorics, Graph Theory & Computing, March 2026

Carnegie Mellon University

Dedication

We sincerely dedicate our work to the life of Natasha Parker, without whom this project would not exist. While we never had the honor of meeting Natasha, we feel, through our peers and mentors, the profound impact she had on those around her. May her memory be a blessing.

We are deeply grateful to Natasha Parker's family, whose generosity and support made our attendance at this conference possible.

Table of Contents

Two-Color Ramsey Numbers

Two-Color Ramsey Numbers

Definition

$R(s,t)$ is the smallest n such that every 2-coloring of the edges of K_n contains a monochromatic K_s in the first color *or* a monochromatic K_t in the second.

Two-Color Ramsey Numbers

Definition

$R(s, t)$ is the smallest n such that every 2-coloring of the edges of K_n contains a monochromatic K_s in the first color *or* a monochromatic K_t in the second.

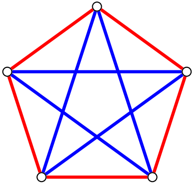
We can think of 2-colorings of K_n as simple graphs: blue edges are **edges**, and red edges are **non-edges**.

Two-Color Ramsey Numbers

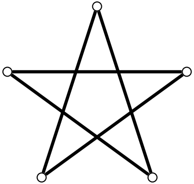
Definition

$R(s,t)$ is the smallest n such that every 2-coloring of the edges of K_n contains a monochromatic K_s in the first color *or* a monochromatic K_t in the second.

We can think of 2-colorings of K_n as simple graphs: blue edges are **edges**, and red edges are **non-edges**.



2-coloring of K_5



simple graph (blue kept as edges)

Two-Color Ramsey Numbers: $R(s, t)$-good

Two-Color Ramsey Numbers: $R(s, t)$-good

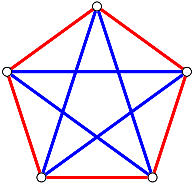
Definition

An n-vertex graph is $R(s, t)$-**good** if it contains no clique K_s and no independent set I_t.

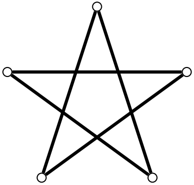
Two-Color Ramsey Numbers: $R(s,t)$-good

Definition

An n-vertex graph is $R(s,t)$-**good** if it contains no clique K_s and no independent set I_t.



no blue K_3 and no red K_3



no K_3 and no I_3

Therefore this is an $R(3,3)$-good graph on 5 vertices, so it **witnesses** $R(3,3) > 5$ (in fact $R(3,3) = 6$).

Strongly Regular Graphs

Strongly Regular Graphs

Regularity

A graph is k-**regular** if every vertex has exactly k neighbors.

Strongly Regular Graphs

Regularity

A graph is k-**regular** if every vertex has exactly k neighbors.

Strong Regularity: $\mathrm{srg}(v, k, \lambda, \mu)$

A k-regular graph on v vertices is **strongly regular** if every pair has

- exactly λ common neighbors when adjacent,
- exactly μ common neighbors when non-adjacent.

Strongly Regular Graphs

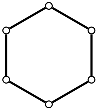
Regularity

A graph is k-**regular** if every vertex has exactly k neighbors.

Strong Regularity: $\mathrm{srg}(v, k, \lambda, \mu)$

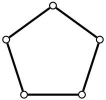
A k-regular graph on v vertices is **strongly regular** if every pair has

- exactly λ common neighbors when adjacent,
- exactly μ common neighbors when non-adjacent.



Regular, not SRG:

C_6 (2-regular; μ varies)



Strongly regular:

$C_5 = \mathrm{srg}(5, 2, 0, 1)$

Carnegie Mellon University

Motivating the Search

Motivating the Search

A Compelling Pattern

The classic diagonal Ramsey numbers are witnessed by strongly regular graphs on exactly $R(k,k) - 1$ vertices:

- $R(3,3) = 6$ is witnessed by $\mathrm{srg}(5,2,0,1)$.
- $R(4,4) = 18$ is witnessed by $\mathrm{srg}(17,8,3,4)$.

A Compelling Pattern

The classic diagonal Ramsey numbers are witnessed by strongly regular graphs on exactly $R(k, k) - 1$ vertices:

- $R(3, 3) = 6$ is witnessed by $\mathrm{srg}(5, 2, 0, 1)$.
- $R(4, 4) = 18$ is witnessed by $\mathrm{srg}(17, 8, 3, 4)$.

Could a similar graph witness $R(5, 5)$?

A Compelling Pattern

The classic diagonal Ramsey numbers are witnessed by strongly regular graphs on exactly $R(k,k)-1$ vertices:

- $R(3,3)=6$ is witnessed by $\mathrm{srg}(5,2,0,1)$.
- $R(4,4)=18$ is witnessed by $\mathrm{srg}(17,8,3,4)$.

Could a similar graph witness $R(5,5)$?

The Target: $v=45$

Within $43 \le R(5,5) \le 46$, the only viable parameter set that could be $R(5,5)$-good is:

$$\mathrm{srg}(45,22,10,11).$$

Identifying such a graph would constructively prove $R(5,5)=46$.

Motivating the Search

A Compelling Pattern

The classic diagonal Ramsey numbers are witnessed by strongly regular graphs on exactly $R(k,k)-1$ vertices:

- $R(3,3)=6$ is witnessed by $\mathrm{srg}(5,2,0,1)$.
- $R(4,4)=18$ is witnessed by $\mathrm{srg}(17,8,3,4)$.

Could a similar graph witness $R(5,5)$?

Goal: find an $R(5,5)$-good graph in $\mathrm{srg}(45,22,10,11)$

The Target: $v=45$

Within $43 \leq R(5,5) \leq 46$, the only viable parameter set that could be $R(5,5)$-good is:

$$\mathrm{srg}(45,22,10,11).$$

Identifying such a graph would constructively prove $R(5,5)=46$.

Harnessing SAT Solvers

Harnessing SAT Solvers

What is a SAT solver?

A **SAT solver** decides whether a Boolean formula is **satisfiable**: can we assign True/False values to variables so that *all* constraints hold?

Harnessing SAT Solvers

What is a SAT solver?

A **SAT solver** decides whether a Boolean formula is **satisfiable**: can we assign True/False values to variables so that *all* constraints hold?

CaDiCaL (CDCL)

We use **CaDiCaL**, a state-of-the-art **CDCL** solver. When a partial assignment leads to a contradiction, it learns a clause explaining the failure, then **backjumps** to skip entire families of similar dead ends.

Harnessing SAT Solvers

What is a SAT solver?

A **SAT solver** decides whether a Boolean formula is **satisfiable**: can we assign True/False values to variables so that *all* constraints hold?

CaDiCaL (CDCL)

We use **CaDiCaL**, a state-of-the-art **CDCL** solver. When a partial assignment leads to a contradiction, it learns a clause explaining the failure, then **backjumps** to skip entire families of similar dead ends.

Constraint in CNF		**DIMACS (plain text)**

$(x_1 \vee \overline{x}_2 \vee x_3) \;\wedge\; (\overline{x}_1 \vee x_4)$	$\Rightarrow$	`p cnf 4 2`
		`1 -2 3 0`
		`-1 4 0`

Encoding idea: translate "graph properties" into Boolean variables and CNF clauses (constraints).

Harnessing SAT Solvers

SAT Solver Example

$$(x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3) \wedge (x_2)$$

SAT solver output: **satisfiable!**

$x_1 = 0,\ x_2 = 1, x_3 = 0$

$$(x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_2) \wedge (x_2)$$

SAT solver output: **unsatisfiable!**

Table of Contents

Encoding $R(5,5)$-Goodness

Encoding $R(5,5)$-Goodness

Variables = edges

For $1 \leq i < j \leq v$, the Boolean variable e_{ij} encodes the edge $\{i,j\}$:

$$e_{ij} = 1 \text{ (edge)} \qquad e_{ij} = 0 \text{ (non-edge)}.$$

Encoding $R(5,5)$-Goodness

Variables = edges

For $1 \leq i < j \leq v$, the Boolean variable e_{ij} encodes the edge $\{i, j\}$:

$$e_{ij} = 1 \text{ (edge)} \qquad e_{ij} = 0 \text{ (non-edge)}.$$

$R(5,5)$-goodness

A graph is $R(5,5)$-good $\iff$ every 5-vertex subset S contains **at least one edge** and **at least one non-edge**.

Encoding $R(5,5)$-Goodness

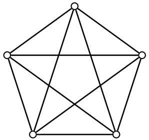
Variables = edges

For $1 \leq i < j \leq v$, the Boolean variable e_{ij} encodes the edge $\{i, j\}$:

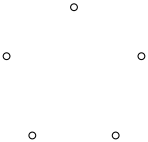
$$e_{ij} = 1 \text{ (edge)} \qquad\qquad e_{ij} = 0 \text{ (non-edge)}.$$

$R(5,5)$-goodness

A graph is $R(5,5)$-good $\iff$ every 5-vertex subset S contains **at least one edge** and **at least one non-edge**.



K_5: all pairs are edges



I_5: no pairs are edges

Encoding $R(5,5)$-Goodness

Encoding $R(5,5)$-Goodness

Fix any 5-vertex subset $S \subseteq [v]$. To enforce the Ramsey condition on S, we add **two clauses**:

Carnegie Mellon University

Encoding $R(5,5)$-Goodness

Fix any 5-vertex subset $S \subseteq [v]$. To enforce the Ramsey condition on S, we add **two clauses**:

Prevent an independent set I_5 in S

At least one edge must appear inside S:

$$\bigvee_{i,j \in S,\ i<j} e_{ij}.$$

Encoding $R(5,5)$-Goodness

Fix any 5-vertex subset $S \subseteq [v]$. To enforce the Ramsey condition on S, we add **two clauses**:

Prevent an independent set I_5 in S

At least one edge must appear inside S:

$$\bigvee_{i,j \in S,\ i<j} e_{ij}.$$

Prevent a clique K_5 in S

At least one non-edge must appear inside S:

$$\bigvee_{i,j \in S,\ i<j} \neg e_{ij}.$$

Since $v = 45$, repeat this pair of clauses $\binom{45}{5}$ times.

Encoding Regularity

Encoding Regularity

Regularity ($k = 22$):
In $\mathrm{srg}(45, 22, 10, 11)$, each vertex has **exactly 22 neighbors** among the other 44 vertices.

Encoding Regularity

Regularity $(k = 22)$**:**
In $\mathrm{srg}(45, 22, 10, 11)$, each vertex has **exactly 22 neighbors** among the other 44 vertices.

KNF: compact cardinality constraints

We write constraints like "*at least t of these variables are true*," then compile them to CNF (e.g., via `knf2cnf`).

Encoding Regularity

Regularity ($k = 22$)**:**
In $\mathrm{srg}(45, 22, 10, 11)$, each vertex has **exactly 22 neighbors** among the other 44 vertices.

KNF: compact cardinality constraints

We write constraints like "*at least t of these variables are true*," then compile them to CNF (e.g., via `knf2cnf`).

Encoding "exactly 22" for vertex i

Require **at least 22 incident edges** and **at least 22 non-edges**:

$$\mathtt{k}\ 22\quad e_{i1}\ \ e_{i2}\ \ \cdots\ \ e_{i,44}$$
$$\mathtt{k}\ 22\quad \neg e_{i1}\ \ \neg e_{i2}\ \ \cdots\ \ \neg e_{i,44}$$

("≥ 22 true" and "≥ 22 false" $\Rightarrow$ exactly 22 true.)

Auxiliary Variables for Common Neighbors

Auxiliary Variables for Common Neighbors

Strong regularity — what we need:
Adjacent pairs share **10** common neighbors; non-adjacent pairs share **11**.

Auxiliary Variables for Common Neighbors

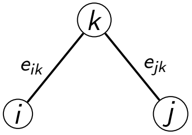
Strong regularity — what we need:
Adjacent pairs share **10** common neighbors; non-adjacent pairs share **11**.

Auxiliary variable for common neighbors

Introduce

$$v_{ijk} \iff (e_{ik} \wedge e_{jk}),$$

so $v_{ijk} = 1$ exactly when k is adjacent to *both* i and j.



$v_{ijk} = 1$ iff both

Carnegie Mellon University

Auxiliary Variables for Common Neighbors

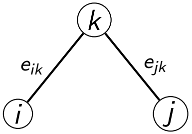
Strong regularity — what we need:
Adjacent pairs share **10** common neighbors; non-adjacent pairs share **11**.

Auxiliary variable for common neighbors

Introduce

$$v_{ijk} \iff (e_{ik} \wedge e_{jk}),$$

so $v_{ijk} = 1$ exactly when k is adjacent to *both* i and j.



$v_{ijk} = 1$ iff both

Key point: Fix a pair $i < j$. There are **43** remaining vertices $k \neq i, j$. Each v_{ijk} flags whether k is a *common neighbor* of i and j, so

$$\sum_{k \neq i,j} v_{ijk} \quad \text{is the number of common neighbors of } i \text{ and } j.$$

Encoding Strong Regularity ($\lambda = 10, \ \mu = 11$)

Encoding Strong Regularity ($\lambda = 10, \ \mu = 11$)

What we need to enforce

Strong regularity in $\mathrm{srg}(45, 22, 10, 11)$ means:

$$e_{ij} = 1 \Rightarrow \sum_{k \neq i,j} v_{ijk} = 10, \qquad e_{ij} = 0 \Rightarrow \sum_{k \neq i,j} v_{ijk} = 11.$$

Encoding Strong Regularity ($\lambda = 10, \ \mu = 11$)

What we need to enforce

Strong regularity in $\mathrm{srg}(45, 22, 10, 11)$ means:

$$e_{ij} = 1 \Rightarrow \sum_{k \neq i,j} v_{ijk} = 10, \qquad e_{ij} = 0 \Rightarrow \sum_{k \neq i,j} v_{ijk} = 11.$$

The key trick: one equation handles both cases

We enforce

$$e_{ij} + \sum_{k \neq i,j} v_{ijk} = 11.$$

- If $e_{ij} = 1$ (adjacent), the sum of v_{ijk} must be 10.
- If $e_{ij} = 0$ (non-adjacent), the sum of v_{ijk} must be 11.

That matches $\lambda = 10$ and $\mu = 11$ exactly.

Encoding Strong Regularity ($\lambda = 10, \ \mu = 11$)

What we need to enforce

Strong regularity in $\mathrm{srg}(45, 22, 10, 11)$ means:

$$e_{ij} = 1 \Rightarrow \sum_{k \neq i,j} v_{ijk} = 10, \qquad e_{ij} = 0 \Rightarrow \sum_{k \neq i,j} v_{ijk} = 11.$$

The key trick: one equation handles both cases

We enforce

$$e_{ij} + \sum_{k \neq i,j} v_{ijk} = 11.$$

- If $e_{ij} = 1$ (adjacent), the sum of v_{ijk} must be 10.
- If $e_{ij} = 0$ (non-adjacent), the sum of v_{ijk} must be 11.

KNF form

We write exactly 11 of the variables e_{ij} plus the 43 values v_{ijk}:

$$\mathtt{k}\ 11 \quad e_{ij} \quad v_{ij,k_1} \quad v_{ij,k_2} \quad \cdots \quad v_{ij,k_{43}}$$

$$\mathtt{k}\ 33 \quad \neg e_{ij} \quad \neg v_{ij,k_1} \quad \neg v_{ij,k_2} \quad \cdots \quad \neg v_{ij,k_{43}}$$

Canonical Form

Canonical Form

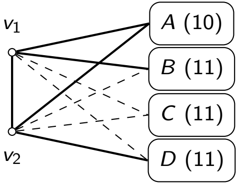
Why a canonical form?

The current constraints ignore vertex labels. A canonical form fixes one labeling around an anchored edge, so the solver searches *structure*, not isomorphic renamings.

Canonical Form

Why a canonical form?

The current constraints ignore vertex labels. A canonical form fixes one labeling around an anchored edge, so the solver searches *structure*, not isomorphic renamings.



solid = edge, dashed = non-edge

Canonical Form

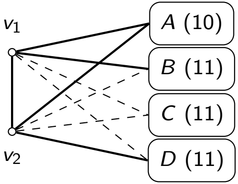
Why a canonical form?

The current constraints ignore vertex labels. A canonical form fixes one labeling around an anchored edge, so the solver searches *structure*, not isomorphic renamings.

Anchor edge. Set $e_{1,2} = 1$ and distinguish $v_1 \sim v_2$.

The remaining vertices split into four **clusters** forced by $(k, \lambda) = (22, 10)$:

$$\begin{aligned} A &: \quad \text{adjacent to both } v_1, v_2 & (10) \\ B &: \quad \text{adjacent to } v_1 \text{ not } v_2 & (11) \\ D &: \quad \text{adjacent to } v_2 \text{ not } v_1 & (11) \\ C &: \quad \text{adjacent to neither } v_1 \text{ nor } v_2 & (11) \end{aligned}$$



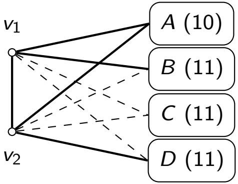
solid = edge, dashed = non-edge

Canonical Form

Canonical Form

The benefit: A is constrained

Every vertex in A sees both v_1 and v_2. Any triangle inside A would form a K_5 with $\{v_1, v_2\}$.

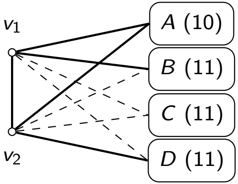


solid = edge, dashed = non-edge

Canonical Form

The benefit: A is constrained

Every vertex in A sees both v_1 and v_2. Any triangle inside A would form a K_5 with $\{v_1, v_2\}$.



solid = edge, dashed = non-edge

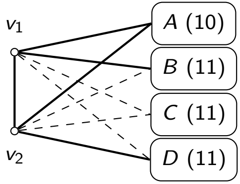
Enumerating A

So the subgraph induced by A must be $R(3, 5)$-**good**. There are exactly **313 non-isomorphic** such graphs on 10 vertices, so we split into **313 cases**.

Canonical Form

The benefit: A is constrained

Every vertex in A sees both v_1 and v_2. Any triangle inside A would form a K_5 with $\{v_1, v_2\}$.



solid = edge, dashed = non-edge

Enumerating A

So the subgraph induced by A must be $R(3,5)$-**good**. There are exactly **313 non-isomorphic** such graphs on 10 vertices, so we split into **313 cases**.

What remains free

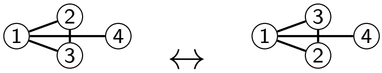
Beyond the forced adjacencies in the diagram (and the chosen case for A), all other edges **within and between** the clusters A, B, C, D are free variables.

Symmetry Breaking

Symmetry Breaking

What is symmetry?

A symmetry is a relabeling of vertices that preserves all edges/non-edges. To a SAT solver, that relabeling changes variable names (like e_{ij}), so it can look like a new case.

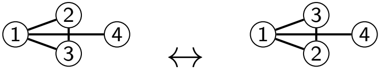


same graph, different labels ($2 \leftrightarrow 3$)

Symmetry Breaking

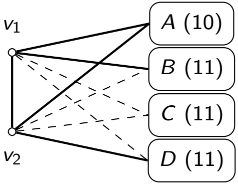
What is symmetry?

A symmetry is a relabeling of vertices that preserves all edges/non-edges. To a SAT solver, that relabeling changes variable names (like e_{ij}), so it can look like a new case.



same graph, different labels ($2 \leftrightarrow 3$)

Where it appears here. After fixing the canonical form, the vertices *within* each cluster B, C, and D are still interchangeable: permuting their labels produces an isomorphic instance.

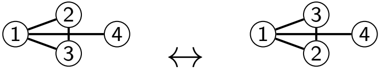


solid = edge, dashed = non-edge

Symmetry Breaking

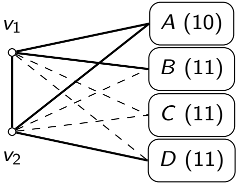
What is symmetry?

A symmetry is a relabeling of vertices that preserves all edges/non-edges. To a SAT solver, that relabeling changes variable names (like e_{ij}), so it can look like a new case.



same graph, different labels $(2 \leftrightarrow 3)$

Where it appears here. After fixing the canonical form, the vertices *within* each cluster B, C, and D are still interchangeable: permuting their labels produces an isomorphic instance.



solid = edge, dashed = non-edge

That is up to $(11!)^3 \approx 6 \times 10^{22}$ redundant labelings of the same structure.

Symmetry Breaking

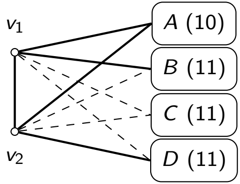
Symmetry Breaking

Adjacency vectors

With A fixed, each vertex $x \in B \cup C \cup D$ has a 10-bit adjacency vector to A:

$$\sigma_A(x) := (e_{x,a_1}, \ldots, e_{x,a_{10}}) \in \{0,1\}^{10}.$$

These vectors let us compare vertices by structure (not by name).



solid = edge, dashed = non-edge

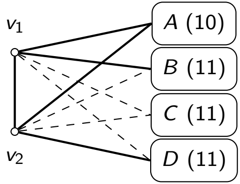
Symmetry Breaking

Adjacency vectors

With A fixed, each vertex $x \in B \cup C \cup D$ has a 10-bit adjacency vector to A:

$$\sigma_A(x) := (e_{x,a_1}, \ldots, e_{x,a_{10}}) \in \{0,1\}^{10}.$$

These vectors let us compare vertices by structure (not by name).



solid = edge, dashed = non-edge

Lex-ordering within each cluster

Within each class (say $B = \{b_1, \ldots, b_{11}\}$), we enforce a canonical order:

$$\sigma_A(b_1) \le_{\text{lex}} \sigma_A(b_2) \le_{\text{lex}} \cdots \le_{\text{lex}} \sigma_A(b_{11})$$

This removes the 11! relabelings inside each of B, C, D (up to a $(11!)^3$ reduction).

Table of Contents

Introduction
Ramsey Numbers
Strongly Regular Graphs
SAT Solvers
Encoding Scheme
$R(5, 5)$-goodness
Strong Regularity
Canonical Form
Symmetry-Breaking
Improved $R(5,5)$-Goodness
Alternate Encoding
Computation Improvements
Conclusion

Improved R(5,5)-goodness

Improved R(5,5)-goodness

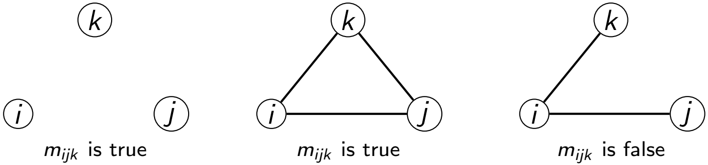
Auxiliary variables for homogeneous triangles

m_{ijk} is true if the induced subgraph of vertices i, j, k forms a independent set or clique of size 3

Improved R(5,5)-goodness

Auxiliary variables for homogeneous triangles

m_{ijk} is true if the induced subgraph of vertices i, j, k forms a independent set or clique of size 3



Improved R(5,5)-goodness

Improved R(5,5)-goodness

Enforcing auxiliary variables in encoding

Encoding each m_{ijk} takes two clauses in our CNF formula

- $m_{ijk} \lor e_{ij} \lor e_{ik} \lor e_{jk}$
- $m_{ijk} \lor \neg e_{ij} \lor \neg e_{ik} \lor \neg e_{jk}$

Improved R(5,5)-goodness

Enforcing auxiliary variables in encoding

Encoding each m_{ijk} takes two clauses in our CNF formula

- $m_{ijk} \lor e_{ij} \lor e_{ik} \lor e_{jk}$
- $m_{ijk} \lor \neg e_{ij} \lor \neg e_{ik} \lor \neg e_{jk}$

Logically equivalent to:

- $\neg e_{ij} \land \neg e_{ik} \land \neg e_{jk} \to m_{ijk}$
- $e_{ij} \land e_{ik} \land e_{jk} \to m_{ijk}$

Improved R(5,5)-goodness

Enforcing auxiliary variables in encoding

Encoding each m_{ijk} takes two clauses in our CNF formula

- $m_{ijk} \vee e_{ij} \vee e_{ik} \vee e_{jk}$
- $m_{ijk} \vee \neg e_{ij} \vee \neg e_{ik} \vee \neg e_{jk}$

Logically equivalent to:

- $\neg e_{ij} \wedge \neg e_{ik} \wedge \neg e_{jk} \rightarrow m_{ijk}$
- $e_{ij} \wedge e_{ik} \wedge e_{jk} \rightarrow m_{ijk}$

We only encode the forward implication. This doesn't affect correctness of SAT encoding, and empirically runs faster than encoding bi-implication.

Improved R(5,5)-goodness

Improved R(5,5)-goodness

Using auxiliary variables to check R(5,5)-goodness

For an induced subgraph of size 5, we pick 5 triangles in it, and check respective auxiliary variables for at least one false. One clause checks for both non-existence of I_5 and K_5

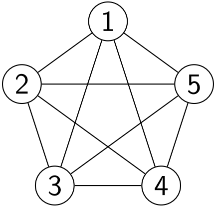
Improved R(5,5)-goodness

Using auxiliary variables to check R(5,5)-goodness

For an induced subgraph of size 5, we pick 5 triangles in it, and check respective auxiliary variables for at least one false. One clause checks for both non-existence of I_5 and K_5

Example clause using auxiliary variables:

$$\neg m_{134} \vee \neg m_{245} \vee \neg m_{135} \vee \neg m_{124} \vee \neg m_{235}$$



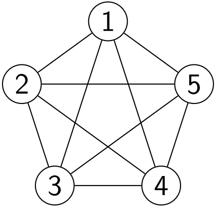
Improved R(5,5)-goodness

Using auxiliary variables to check R(5,5)-goodness

For an induced subgraph of size 5, we pick 5 triangles in it, and check respective auxiliary variables for at least one false. One clause checks for both non-existence of I_5 and K_5

Example clause using auxiliary variables:

$\neg m_{134} \vee \neg m_{245} \vee \neg m_{135} \vee \neg m_{124} \vee \neg m_{235}$



5 triangles cover all 10 edges, and are connected by overlapping edges. Ensures clause is unsatisfiable iff I_5 or K_5 exists

Theoretical and Empirical Improvements

Theoretical and Empirical Improvements

Old encoding scheme:

- 1 clause of 10 variables to check for I_5
- 1 clause of 10 variables to check for K_5
- $2 * \binom{45}{5} = 2443518$ clauses in total

Theoretical and Empirical Improvements

Old encoding scheme:

- 1 clause of 10 variables to check for I_5
- 1 clause of 10 variables to check for K_5
- $2 * \binom{45}{5} = 2443518$ clauses in total

New encoding scheme:

- 2 clauses of 4 variables to enforce every auxiliary variable
- 1 clause of 5 variables to check for I_5 and K_5
- $2 * \binom{45}{3} + \binom{45}{5} = 1250139$ clauses in total

Theoretical and Empirical Improvements

Old encoding scheme:

- 1 clause of 10 variables to check for I_5
- 1 clause of 10 variables to check for K_5
- $2 * \binom{45}{5} = 2443518$ clauses in total

New encoding scheme:

- 2 clauses of 4 variables to enforce every auxiliary variable
- 1 clause of 5 variables to check for I_5 and K_5
- $2 * \binom{45}{3} + \binom{45}{5} = 1250139$ clauses in total

Number of clauses and variables per clause are both cut in half

Theoretical and Empirical Improvements

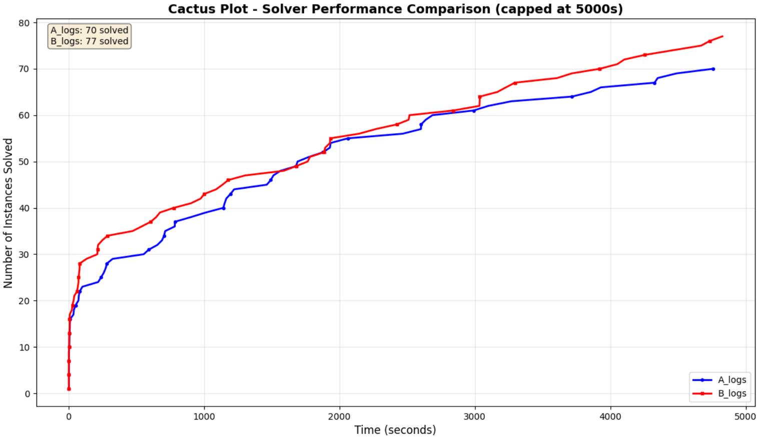


Table of Contents

Conclusion

Conclusion

What we achieved

- **Progress:** proved **141 of 313** hard-coded cases are **UNSAT**.
- **Method:** built a structured SAT pipeline for $\mathrm{srg}(45, 22, 10, 11)$ under $R(5, 5)$-goodness.
- **Speedup:** developed a faster encoding for checking $R(5, 5)$-goodness (K_5/I_5-freeness).

Conclusion

What we achieved

- **Progress:** proved **141 of 313** hard-coded cases are **UNSAT**.
- **Method:** built a structured SAT pipeline for $\mathrm{srg}(45, 22, 10, 11)$ under $R(5, 5)$-goodness.
- **Speedup:** developed a faster encoding for checking $R(5, 5)$-goodness (K_5/I_5-freeness).

If any remaining case is satisfiable: we obtain an $R(5, 5)$-good graph on 45 vertices, hence $R(5, 5) = 46$.

Conclusion

What we achieved

- **Progress:** proved **141 of 313** hard-coded cases are **UNSAT**.
- **Method:** built a structured SAT pipeline for $\mathrm{srg}(45, 22, 10, 11)$ under $R(5, 5)$-goodness.
- **Speedup:** developed a faster encoding for checking $R(5, 5)$-goodness (K_5/I_5-freeness).

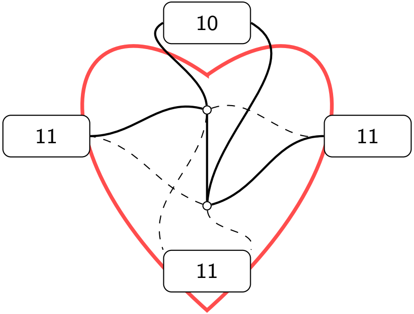
If any remaining case is satisfiable: we obtain an $R(5, 5)$-good graph on 45 vertices, hence $R(5, 5) = 46$.

Future Directions

More scaffolding: impose additional canonical "skeleton" structure on the free clusters B, C, D—extra constraints that preserve existence but cut isomorphic search.

Acknowledgments

We thank *Professor John Mackey* and *Zachary Battleman* for their mentorship and guidance throughout this project. We are grateful for their time, feedback, and continued support.



solid = edge   dashed = non-edge