Combinatorics Using Computational Methods

Derrick Stolee

University of Nebraska–Lincoln
s-dstolee1@math.unl.edu
http://www.math.unl.edu/~s-dstolee1/

March 13, 2012
Dissertation Defense

Supported by NSF grants DMS-0354008, DMS-0914815, and CCF-0916525,
and a University of Nebraska Presidential Fellowship.
Advisors and Committee

Stephen Hartke  
Mathematics

Vinod Variyam  
Computer Science and Engineering

Jamie Radcliffe  
Mathematics

Stephen Scott  
CSE

Christina Falci  
Sociology
Thanks to...

Katie Stolee

Eric Allender, Pranav Anand, David Barrington, Brian Bockleman, Chris Bourke, Jane Butterfield, James Carraher, Henry Escuadro, Michael Ferrara, Lance Fortnow, Brady Garvin, Raluca Gera, Joe Geisbauer, Ellen Gethner, Steve Goddard, Adam S. Jobson, Travis Johnston, André Kézdy, Elizabeth Kupin, Timothy D. LeSaulnier, Jared Nishikawa, Kevin G. Milans, Andrew Ray, Ben Reiniger, Tyler Seacrest, Hannah (Kolb) Spinoza, Brendon Stanton, David Swanson, Raghunath Tewari, Judy Walker, Derek Weitzel, Paul S. Wenger, Douglas B. West, Zahava Wilstein, Matthew Yancey, and

all UNL Math or CSE graduate students, staff, and faculty.

Full Acknowledgements available on my web page.
Computational Combinatorics

Pure Combinatorics

Algorithms and Computation
Computational Combinatorics

Pure Combinatorics

Problem

Algorithms and Computation
Computational Combinatorics

Problem

Pure Combinatorics

Examples

Algorithms and Computation
Computational Combinatorics

Pure Combinatorics

Problem

Structure

Examples

Algorithms and Computation
Computational Combinatorics
Computational Combinatorics

Pure Combinatorics

Problem
Structure
Efficiency
Perspective
Examples

Algorithms and Computation
Computational Combinatorics

Pure Combinatorics

Problem
Structure
Efficiency

Theorems
Perspective
Examples

Algorithms and Computation
Computational Combinatorics

Pure Combinatorics

Problem
Structure
Efficiency

Algorithms and Computation

Computational Combinatorics

Theorems
Perspective
Examples
The Goal

Determine if certain **combinatorial objects** exist with given **structural** or **extremal** properties.
The Goal

Determine if certain **combinatorial objects** exist with given **structural** or **extremal** properties.

**Examples:**

1. Is there a **projective plane** of order 10?
2. When do **strongly regular graphs** exist?
3. How many **Steiner triple systems** are there of order 19?
The Goal

Determine if certain **combinatorial objects** exist with given **structural** or **extremal** properties.

**Examples:**

1. Is there a **projective plane** of order 10?
   (Lam, Thiel, Swiercz, 1989)

2. When do **strongly regular graphs** exist?
   (Spence 2000, Coolsaet, Degraer, Spence 2006, many others)

3. How many **Steiner triple systems** are there of order 19?
   (Kaski, Östergård, 2004)
Problems Tackled in This Thesis

1. Which numbers are representable as the number of chains in a width-two poset? (with Kupin, Reiniger)

2. Which colorings of $\{1, \ldots, n\}$ avoid monochromatic progressions? (with Jobson, K´ezdy)

3. How many edges can exist in a graph with $p$ perfect matchings? (with Hartke, West, Yancey)

4. What graphs are uniquely $K_r$-saturated? (with Hartke)
Problems Tackled in This Thesis

1. Which numbers are representable as the number of chains in a width-two poset?  
   (with Kupin, Reiniger)  
   Chapter 4

2. Which colorings of $\{1, \ldots, n\}$ avoid monochromatic progressions?  
   (with Jobson, Kézdy)  
   Chapter 5

3. How many edges can exist in a graph with $p$ perfect matchings?  
   (with Hartke, West, Yancey)  
   Chapter 9

4. What graphs are uniquely $K_r$-saturated?  
   (with Hartke)  
   Chapter 11
Problems Tackled in This Thesis

1. Which numbers are representable as the number of chains in a width-two poset? (with Kupin, Reiniger) Chapter 4

2. Which colorings of \( \{1, \ldots, n\} \) avoid monochromatic progressions? (with Jobson, Kézdy) Chapter 5

3. How many edges can exist in a graph with \( p \) perfect matchings? (with Hartke, West, Yancey) Chapter 9

4. What graphs are uniquely \( K_r \)-saturated? (with Hartke) Chapter 11
Main Technique: Combinatorial Search

**Goal:** Determine if certain combinatorial objects exist with given structural or extremal properties.

**Idea:** Build objects *piece-by-piece* from *base examples* to enumerate all desired examples of a given order.
Main Technique: Combinatorial Search

**Goal:** Determine if certain combinatorial objects exist with given structural or extremal properties.

**Idea:** Build objects *piece-by-piece* from *base examples* to enumerate all desired examples of a given order.

*Most interesting properties are invariant under isomorphism.*
A **graph** $G$ of **order** $n$ is composed of a set $V(G)$ of $n$ vertices and a set $E(G)$ of edges, where the edges are unordered pairs of vertices.
Combinatorial Object: Graphs

A graph \( G \) of order \( n \) is composed of a set \( V(G) \) of \( n \) vertices and a set \( E(G) \) of edges, where the edges are unordered pairs of vertices.
Combinatorial Object: Graphs

An **isomorphism** between $G_1$ and $G_2$ is a bijection from $V(G_1)$ to $V(G_2)$ that induces a bijection from $E(G_1)$ to $E(G_2)$. 
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K}_n$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K}_n$ by adding edges.
Two Techniques for Isomorphs

1. Canonical Deletion (McKay 1998)
   - Removes all isomorphs.
   - Not known how to integrate with constraint propagation.
   - High cost per object.

2. Orbital Branching (Ostrowski, Linderoth, Rossi, Smriglio 2007)
   - Removes some, but not all isomorphs.
   - Naturally integrates with constraint propagation.
   - Low cost per object.
Two Techniques for Isomorphs

1. **Canonical Deletion** *(McKay 1998)*
   - Removes all isomorphs.
   - Not known how to integrate with constraint propagation.
   - High cost per object.

   **Overview in Chapter 6**

2. **Orbital Branching** *(Ostrowski, Linderoth, Rossi, Smriglio 2007)*
   - Removes some, but not all isomorphs.
   - Naturally integrates with constraint propagation.
   - Low cost per object.

   **Overview in Chapter 10**
Search by Augmentations
Search by Augmentations
Search by Augmentations
Search by Augmentations
Search by Augmentations
Search by Augmentations
Search by Augmentations
Search by Augmentations
Implementation

My TreeSearch library enables parallelization in the Condor scheduler.

Executes on the Open Science Grid, a collection of supercomputers around the country.
Problems Tackled in This Thesis

1. Which numbers are representable as the number of chains in a width-two poset?
   (with Kupin, Reiniger) Chapter 4

2. Which colorings of \{1, \ldots, n\} avoid monochromatic progressions?
   (with Jobson, Kézdy) Chapter 5

3. How many edges can exist in a graph with \(p\) perfect matchings?
   (with Hartke, West, Yancey) Chapter 9

4. What graphs are uniquely \(K_r\)-saturated?
   (with Hartke) Chapter 11
Problems Tackled in This Thesis

1. Which numbers are representable as the number of chains in a width-two poset?
   (with Kupin, Reiniger)  Chapter 4

2. Which colorings of \{1, \ldots, n\} avoid monochromatic progressions?
   (with Jobson, Kézdy)  Chapter 5

3. How many edges can exist in a graph with \(p\) perfect matchings?
   (with Hartke, West, Yancey)  Chapter 9

4. What graphs are uniquely \(K_r\)-saturated?
   (with Hartke)  Chapter 11
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.
A **perfect matching** is a set of edges which cover each vertex exactly once. 

$\Phi(G)$ is the number of perfect matchings in the graph $G$. 

- ![Diagram of a graph with perfect matchings]
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.

$\Phi(G)$ is the number of perfect matchings in the graph $G$. 
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once. 

\( \Phi(G) \) is the number of perfect matchings in the graph \( G \).

\[ \Phi(G) = 3 \]

8 edges
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.

\[ \Phi(G) = 3 \]
\[ 8 \text{ edges} \]

11 edges
A **perfect matching** is a set of edges which cover each vertex exactly once.

\[ \Phi(G) = 3 \]

8 edges

\[ \Phi(G) = 11 \]

11 edges
A **perfect matching** is a set of edges which cover each vertex exactly once.

\[ \Phi(G) = 3 \]

8 edges

\[ \Phi(G) = 11 \]

11 edges
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.

\[ \Phi(G) = 3 \quad \text{8 edges} \]

\[ \Phi(G) = 3 \quad \text{11 edges} \]
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.

**Question (Dudek, Schmitt, 2010)** What is the maximum number of edges in a graph with exactly $n$ vertices and $p$ perfect matchings?

**Definition** Let $n$ be an even number and fix $p \geq 1$.

$$f(n, p) = \max\{ |E(G)| : |V(G)| = n, \Phi(G) = p \}.$$  

Graphs attaining this number of edges are **$p$-extremal**.
Hetyei’s Theorem

Theorem (Hetyei’s Theorem, 1986) For all even \( n \geq 2 \),

\[
f(n, 1) = \frac{n^2}{4}.
\]
Hetyei’s Theorem

Theorem (Hetyei’s Theorem, 1986) For all even $n \geq 2$, $f(n, 1) = \frac{n^2}{4}$. 

\[
f(n, 1) = \frac{n^2}{4}.
\]
Hetyei’s Theorem

Theorem (Hetyei’s Theorem, 1986) For all even $n \geq 2$,

$$f(n, 1) = \frac{n^2}{4}.$$
Hetyei’s Theorem

Theorem (Hetyei’s Theorem, 1986) For all even $n \geq 2$,

$$f(n, 1) = \frac{n^2}{4}.$$
The Form of $f(n, p)$

**Theorem (Dudek, Schmitt, 2010)** For each $p$, there exist constants $n_p, c_p$ so that for all $n \geq n_p$,

$$f(n, p) = \frac{n^2}{4} + c_p.$$
The Form of $f(n, p)$

**Theorem (Dudek, Schmitt, 2010)** For each $p$, there exist constants $n_p, c_p$ so that for all $n \geq n_p$,

$$f(n, p) = \frac{n^2}{4} + c_p.$$ 

<table>
<thead>
<tr>
<th>$p$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Dudek, Schmitt, 2010
**Structure Theorem**

**Theorem (Hartke, Stolee, West, Yancey, 2011)** For a fixed $p$, every graph $G$ with $n$ vertices, $p$ perfect matchings, and $f(n, p) = \frac{n^2}{4} + c_p$ edges is composed of a finite list of **fundamental graphs** combined in specified ways.

Proof involves several classic structure theorems from matching theory in an extremal setting.
Structure Theorem

**Theorem (Hartke, Stolee, West, Yancey, 2011)** For a fixed $p$, every graph $G$ with $n$ vertices, $p$ perfect matchings, and $f(n, p) = \frac{n^2}{4} + c_p$ edges is composed of a finite list of **fundamental graphs** combined in specified ways.

Proof involves several classic structure theorems from matching theory in an extremal setting.

For $p \leq 10$, the graphs have order at most 12.
Structure Theorem

**Theorem (Hartke, Stolee, West, Yancey, 2011)** For a fixed $p$, every graph $G$ with $n$ vertices, $p$ perfect matchings, and $f(n, p) = \frac{n^2}{4} + c_p$ edges is composed of a finite list of fundamental graphs combined in specified ways.

Proof involves several classic structure theorems from matching theory in an extremal setting.

For $p \leq 10$, the graphs have order at most 12.

Using standard software (McKay’s *geng*) we found the graphs and computed $c_p$. 
Fundamental Graphs for $2 \leq p \leq 10$
### $c_p$ for small $p$

<table>
<thead>
<tr>
<th>$p$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>Dudek, Schmitt 2010</td>
<td>HSWY 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### $c_p$ for small $p$

<table>
<thead>
<tr>
<th>$p$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>Dudek, Schmitt 2010</td>
<td>HSWY 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Q:** Is $c_p$ monotone in $p$?
Without more involved computational methods, brute force methods (such as \textit{geng}) cannot go farther.
Structural Theorem, Redux

Without more involved computational methods, brute force methods (such as *geng*) cannot go farther.

The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.
Structural Theorem, Redux

Without more involved computational methods, brute force methods (such as *geng*) cannot go farther.

The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.
Structural Theorem, Redux

Without more involved computational methods, brute force methods (such as *geng*) cannot go farther.

The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using *ear augmentations*. 
Structural Theorem, Redux

Without more involved computational methods, brute force methods (such as \textit{geng}) cannot go farther.

The \textbf{Lovász Two Ear Theorem (1983)} provides a way to build fundamental graphs using \textit{ear augmentations}. 

\begin{figure}
    \centering
    \begin{tikzpicture}
        \node (A) at (0,0) [circle, fill=black] {};
        \node (B) at (1,0) [circle, fill=black] {};
        \node (C) at (1,1) [circle, fill=black] {};
        \node (D) at (0,1) [circle, fill=black] {};
        \draw (A) -- (B) -- (C) -- (D) -- (A);
        \draw [dashed, red] (A) -- (D);
    \end{tikzpicture}
    \begin{tikzpicture}
        \node (A) at (0,0) [circle, fill=black] {};
        \node (B) at (1,0) [circle, fill=black] {};
        \node (C) at (1,1) [circle, fill=black] {};
        \node (D) at (0,1) [circle, fill=black] {};
        \node (E) at (2,0) [circle, fill=black] {};
        \node (F) at (3,0) [circle, fill=black] {};
        \node (G) at (3,1) [circle, fill=black] {};
        \node (H) at (2,1) [circle, fill=black] {};
        \draw (A) -- (B) -- (C) -- (D) -- (A);
        \draw (E) -- (F) -- (G) -- (H) -- (E);
        \draw [dashed, red] (C) -- (D);
    \end{tikzpicture}
\end{figure}
Structural Theorem, Redux

Without more involved computational methods, brute force methods (such as \textit{geng}) cannot go farther.

The \textbf{Lovász Two Ear Theorem (1983)} provides a way to build fundamental graphs using \textit{ear augmentations}.

\begin{center}
\includegraphics[width=\textwidth]{ear_augmentations.png}
\end{center}
Structural Theorem, Redux

Without more involved computational methods, brute force methods (such as *geng*) cannot go farther.

The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.
Computational Method

Developed a computational method from:

1. **Augmentations**: Lovász Two Ear Theorem.

2. **Isomorphs**: Canonical Deletion.

3. **Pruning**: Developed new structural and extremal theorems.
Computational Method

Developed a computational method from:

1. **Augmentations**: Lovász Two Ear Theorem.

2. **Isomorphs**: Canonical Deletion. 

3. **Pruning**: Developed new structural and extremal theorems.

**Before**: Stuck at $p \leq 10$ when searching on most 12 vertices.
Computational Method

Developed a computational method from:

1. **Augmentations**: Lovász Two Ear Theorem.

2. **Isomorphs**: Canonical Deletion.

3. **Pruning**: Developed new structural and extremal theorems.

**Before**: Stuck at $p \leq 10$ when searching on most 12 vertices.

**Now**: Found graphs for all $p \leq 27$ on up to 22 vertices.
Fundamental Graphs for $11 \leq p \leq 27$
### $c_p$ for small $p$

<table>
<thead>
<tr>
<th>$p$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

H  Dudek, Schmitt, 2010  HSWY, 2011

<table>
<thead>
<tr>
<th>$p$</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Stolee, 2011

<table>
<thead>
<tr>
<th>$p$</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Stolee, 2011
### $c_p$ for small $p$

<table>
<thead>
<tr>
<th>$p$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

H: Dudek, Schmitt, 2010  
HSWY, 2011  

<table>
<thead>
<tr>
<th>$p$</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Stolee, 2011

<table>
<thead>
<tr>
<th>$p$</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Stolee, 2011

$c_p$ not monotonic in $p$!

Blue numbers match conjectured upper bound.
Uniquely $K_r$-Saturated Graphs

Problems Tackled in This Thesis

1. Which numbers are representable as the number of chains in a width-two poset?
   (with Kupin, Reiniger) Chapter 4

2. Which colorings of $\{1, \ldots, n\}$ avoid monochromatic progressions?
   (with Jobson, Kézdy) Chapter 5

3. How many edges can exist in a graph with $p$ perfect matchings?
   (with Hartke, West, Yancey) Chapter 9

4. What graphs are uniquely $K_r$-saturated?
   (with Hartke) Chapter 11
Problems Tackled in This Thesis

1. Which numbers are representable as the number of chains in a width-two poset?
   (with Kupin, Reiniger) Chapter 4

2. Which colorings of \{1, \ldots, n\} avoid monochromatic progressions?
   (with Jobson, Kézdy) Chapter 5

3. How many edges can exist in a graph with \( p \) perfect matchings?
   (with Hartke, West, Yancey) Chapter 9

4. What graphs are uniquely \( K_r \)-saturated?
   (with Hartke) Chapter 11
**H-Saturated Graphs**

**Definition** A graph $G$ is **$H$-saturated** if

- $G$ does not contain $H$ as a subgraph. (**$H$-free**)
- For every $e \in E(G)$, $G + e$ contains $H$ as a subgraph.

Example: $H = K_3$ where $K_r$ is the **complete graph** on $r$ vertices.
**H-Saturated Graphs**

**Definition**  A graph $G$ is **H-saturated** if

- $G$ does not contain $H$ as a subgraph. (**H-free**)
- For every $e \in E(G)$, $G + e$ contains $H$ as a subgraph.

**Example:** $H = K_3$ where $K_r$ is the **complete graph** on $r$ vertices.
**Definition**  
A graph $G$ is **$H$-saturated** if

- $G$ does not contain $H$ as a subgraph. ($H$-free)
- For every $e \in E(G)$, $G + e$ contains $H$ as a subgraph.

---

**Example:** $H = K_3$ where $K_r$ is the **complete graph** on $r$ vertices.
**H-Saturated Graphs**

**Definition** A graph $G$ is **$H$-saturated** if

- $G$ does not contain $H$ as a subgraph. (**$H$-free**)
- For every $e \in E(G)$, $G + e$ contains $H$ as a subgraph.

Example: $H = K_3$ where $K_r$ is the **complete graph** on $r$ vertices.
Turán’s Theorem

**Theorem (Turán, 1941)** Let $r \geq 3$. If $G$ is $K_r$-saturated on $n$ vertices, then $G$ has **at most** $(1 - \frac{1}{r-1}) \frac{n^2}{2}$ edges (asymptotically).
**Turán’s Theorem**

**Theorem (Turán, 1941)** Let $r \geq 3$. If $G$ is $K_r$-saturated on $n$ vertices, then $G$ has at most \( (1 - \frac{1}{r-1}) \frac{n^2}{2} \) edges (asymptotically).
Turán’s Theorem

Theorem (Turán, 1941) Let $r \geq 3$. If $G$ is $K_r$-saturated on $n$ vertices, then $G$ has at most $(1 - \frac{1}{r-1}) \frac{n^2}{2}$ edges (asymptotically).
Erdős, Hajnal, and Moon

**Theorem (Erdős, Hajnal, Moon, 1964)** Let $r \geq 3$. If $G$ is $K_r$-saturated on $n$ vertices, then $G$ has **at least** \( \binom{r-2}{2} + (r - 2)(n - r + 2) \) edges.
Erdős, Hajnal, and Moon

Theorem (Erdős, Hajnal, Moon, 1964) Let $r \geq 3$. If $G$ is $K_r$-saturated on $n$ vertices, then $G$ has at least $\binom{r-2}{2} + (r-2)(n-r+2)$ edges.
Theorem (Erdős, Hajnal, Moon, 1964) Let $r \geq 3$. If $G$ is $K_r$-saturated on $n$ vertices, then $G$ has at least $\binom{r-2}{2} + (r-2)(n-r+2)$ edges.

Exactly one copy of $K_r$!
Uniquely $H$-Saturated Graphs

The Turán graph has many copies of $K_r$ when an edge is added.

The books have exactly one copy of $K_r$ when an edge is added.
Uniquely $H$-Saturated Graphs

The Turán graph has many copies of $K_r$ when an edge is added.

The books have exactly one copy of $K_r$ when an edge is added.

**Definition**  A graph $G$ is **uniquely $H$-saturated** if $G$ does not contain $H$ as a subgraph and for every edge $e \in \overline{G}$ admits exactly one copy of $H$ in $G + e$.

We consider the case where $H = K_r$ (an $r$-clique).
Uniquely $K_3$-Saturated Graphs

**Lemma (Cooper, Lenz, LeSaulnier, Wenger, West, 2011)**
The uniquely $K_3$-saturated graphs are either **stars** or **Moore graphs** of diameter 2 and girth 5.
Uniquely $K_3$-Saturated Graphs

**Lemma (Cooper, Lenz, LeSaulnier, Wenger, West, 2011)**
The uniquely $K_3$-saturated graphs are either stars or Moore graphs of diameter 2 and girth 5.

**Theorem (Hoffman, Singleton, 1964)** There are a finite number of Moore graphs of diameter 2 and girth 5.
Uniquely $K_3$-Saturated Graphs

Lemma (Cooper, Lenz, LeSaulnier, Wenger, West, 2011)  
The uniquely $K_3$-saturated graphs are either stars or Moore graphs of diameter 2 and girth 5.

Theorem (Hoffman, Singleton, 1964)  
There are a finite number of Moore graphs of diameter 2 and girth 5.

\[
\begin{align*}
C_5 & \quad \text{Petersen} & \quad \text{Hoffman–Singleton} & \quad 57\text{-Regular Order 3250} \\
\end{align*}
\]
Adding a dominating vertex to a uniquely $K_r$-saturated graph creates a uniquely $K_{r+1}$-saturated graph.
Adding a dominating vertex to a uniquely $K_r$-saturated graph creates a uniquely $K_{r+1}$-saturated graph.
Dominating Vertices

Adding a dominating vertex to a uniquely $K_r$-saturated graph creates a uniquely $K_{r+1}$-saturated graph.
Dominating Vertices

Adding a dominating vertex to a uniquely $K_r$-saturated graph creates a uniquely $K_{r+1}$-saturated graph.
**Dominating Vertices**

Adding a dominating vertex to a uniquely $K_r$-saturated graph creates a uniquely $K_{r+1}$-saturated graph.

Call uniquely $K_r$-saturated graphs without a dominating vertex \( r \)-primitive.
A uniquely $K_r$-saturated graph with no dominating vertex is $r$-primitive.
A uniquely $K_r$-saturated graph with no dominating vertex is \textit{r-primitive}.

\textbf{2-primitive} graphs are \textit{empty graphs}.
A uniquely $K_r$-saturated graph with no dominating vertex is $r$-primitive.

2-primitive graphs are empty graphs.

3-primitive graphs are Moore graphs of diameter 2 and girth 5.
A uniquely $K_r$-saturated graph with no dominating vertex is \textit{\textbf{$r$-primitive}}.

For $r \geq 1$, $C_{2r-1}$ is $r$-primitive.

(Collins, Cooper, Kay, Wenger, 2010)
A uniquely $K_r$-saturated graph with no dominating vertex is \( r \)-primitive.

For \( r \geq 1 \), \( C_{2r-1} \) is \( r \)-primitive.

(Collins, Cooper, Kay, Wenger, 2010)
A uniquely $K_r$-saturated graph with no dominating vertex is $r$-primitive.

For $r \geq 1$, $C_{2r-1}$ is $r$-primitive.

(Collins, Cooper, Kay, Wenger, 2010)
A uniquely $K_r$-saturated graph with no dominating vertex is \textit{$r$-primitive}.

For $r \geq 1$, $\overline{C_{2r-1}}$ is \textit{$r$-primitive}.

(Collins, Cooper, Kay, Wenger, 2010)
Uniquely $K_4$-Saturated Graphs

Previously known 4-primitive graphs (Collins, Cooper, Kay, 2010)
Two Questions

1. Fix $r \geq 3$. Are there a finite number of $r$-primitive graphs?
2. Is every $r$-primitive graph regular?
Two Questions

1. Fix \( r \geq 3 \). Are there a finite number of \( r \)-primitive graphs?
Two Questions

1. Fix $r \geq 3$. Are there a finite number of $r$-primitive graphs?

2. Is every $r$-primitive graph regular?
Non-edges are crucial to the structure of $r$-primitive graphs.
Edges and Non-Edges

Non-edges are crucial to the structure of $r$-primitive graphs.
\(K_r\)-Completions

For every non-edge we add, we add a \(K_r\)-completion:

\[ij\] a non-edge if and only if there exists a set \(S \subset [n], |S| = r - 2\), so that \(ia, ja, \text{ and } ab\) are edges for all \(a, b \in S\).
Developed a computational method from:

1. **Augmentations**: $K_r$-Completions.

2. **Isomorphs**: Orbital Branching.  
   
   Ostrowsky *et al.*

3. **Pruning**: Contains $K_r$ or double-completion.
## Exhaustive Search Times

<table>
<thead>
<tr>
<th>$n$</th>
<th>$r = 4$</th>
<th>$r = 5$</th>
<th>$r = 6$</th>
<th>$r = 7$</th>
<th>$r = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.10 s</td>
<td>0.37 s</td>
<td>0.13 s</td>
<td>0.01 s</td>
<td>0.01 s</td>
</tr>
<tr>
<td>11</td>
<td>0.68 s</td>
<td>5.25 s</td>
<td>1.91 s</td>
<td>0.28 s</td>
<td>0.09 s</td>
</tr>
<tr>
<td>12</td>
<td>4.58 s</td>
<td>1.60 m</td>
<td>25.39 s</td>
<td>1.97 s</td>
<td>1.12 s</td>
</tr>
<tr>
<td>13</td>
<td>34.66 s</td>
<td>34.54 m</td>
<td>6.53 m</td>
<td>59.94 s</td>
<td>20.03 s</td>
</tr>
<tr>
<td>14</td>
<td>4.93 m</td>
<td>10.39 h</td>
<td>5.13 h</td>
<td>20.66 m</td>
<td>2.71 m</td>
</tr>
<tr>
<td>15</td>
<td>40.59 m</td>
<td>23.49 d</td>
<td>10.08 d</td>
<td>12.28 h</td>
<td>1.22 h</td>
</tr>
<tr>
<td>16</td>
<td>6.34 h</td>
<td>1.58 y</td>
<td>1.74 y</td>
<td>34.53 d</td>
<td>1.88 d</td>
</tr>
<tr>
<td>17</td>
<td>3.44 d</td>
<td></td>
<td></td>
<td>8.76 y</td>
<td>115.69 d</td>
</tr>
<tr>
<td>18</td>
<td>53.01 d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>2.01 y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>45.11 y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total CPU times using Open Science Grid.
<table>
<thead>
<tr>
<th>$n \setminus r$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n \ r</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>-------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Empty graphs
<table>
<thead>
<tr>
<th>$n \setminus r$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>9</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>10</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>11</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>12</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>13</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>14</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>15</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>16</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>17</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>18</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>19</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

- Empty graphs
- Cycle complements
<table>
<thead>
<tr>
<th>$n \setminus r$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>9</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>10</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>11</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>12</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>13</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>14</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>15</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>16</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>17</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>18</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>19</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

- Empty graphs
- Cycle complements
- Old examples
4-Primitive Graphs

$n = 13$

$G_{13}^{(A)}$

Paley(13)
5-Primitive Graph

\[ n = 16 : G_{16}^{(A)} \]
5-Primitive Graph

\[ n = 16 : G_{16}^{(A)} \]
5-Primitive Graph

$n = 16 : G_{16}^{(A)}$

Not all $r$-primitive graphs are regular!
7-Primitive Graph

\[ n = 17 : G_{17}^{(A)} \]
7-Primitive Graph

\[ n = 17 : G_{17}^{(A)} \]
Let $\Gamma$ be a group and $S \subseteq \Gamma$ a set of generators.

The undirected **Cayley graph** $C(\Gamma, S)$ has vertex set $\Gamma$ and for all $a \in \Gamma$ and $b \in S$, there is an edge between $a$ and $ab$. 
Let $\Gamma$ be a group and $S \subseteq \Gamma$ a set of generators.

The undirected \textbf{Cayley graph} $C(\Gamma, S)$ has vertex set $\Gamma$ and for all $a \in \Gamma$ and $b \in S$, there is an edge between $a$ and $ab$.

The \textbf{Cayley complement} $\overline{C}(\Gamma, S)$ is the complement of $C(\Gamma, S)$. 
Let $\Gamma$ be a group and $S \subseteq \Gamma$ a set of generators.

The undirected **Cayley graph** $C(\Gamma, S)$ has vertex set $\Gamma$ and for all $a \in \Gamma$ and $b \in S$, there is an edge between $a$ and $ab$.

The **Cayley complement** $\overline{C}(\Gamma, S)$ is the complement of $C(\Gamma, S)$.

For $r \geq 1$, $\overline{C}(\mathbb{Z}_{2r-1}, \{1\}) \cong \overline{C_{2r-1}}$ is $r$-primitive.
### Two or Three Generators

<table>
<thead>
<tr>
<th>$S$</th>
<th>$r$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${1, 4}$</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>${1, 6}$</td>
<td>16</td>
<td>37</td>
</tr>
<tr>
<td>${1, 8}$</td>
<td>29</td>
<td>65</td>
</tr>
<tr>
<td>${1, 10}$</td>
<td>46</td>
<td>101</td>
</tr>
<tr>
<td>${1, 12}$</td>
<td>67</td>
<td>145</td>
</tr>
</tbody>
</table>

$g = 2$

<table>
<thead>
<tr>
<th>$S$</th>
<th>$r$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${1, 5, 6}$</td>
<td>9</td>
<td>31</td>
</tr>
<tr>
<td>${1, 8, 9}$</td>
<td>22</td>
<td>73</td>
</tr>
<tr>
<td>${1, 11, 12}$</td>
<td>41</td>
<td>133</td>
</tr>
<tr>
<td>${1, 14, 15}$</td>
<td>66</td>
<td>211</td>
</tr>
<tr>
<td>${1, 17, 18}$</td>
<td>97</td>
<td>307</td>
</tr>
</tbody>
</table>

$g = 3$
Infinite Families

Conjecture (Hartke, Stolee, 2012) Let $t \geq 1$,

$$n = 4t^2 + 1, \quad \text{and} \quad r = 2t^2 - t + 1.$$ 

The Cayley complement $\overline{C}(\mathbb{Z}_n, \{1, 2t\})$ is $r$-primitive.

Conjecture (Hartke, Stolee, 2012) Let $t \geq 1$,

$$n = 9t^2 - 3t + 1 \quad \text{and} \quad r = 3t^2 - 2t + 1.$$ 

The Cayley complement $\overline{C}(\mathbb{Z}_n, \{1, 3t - 1, 3t\})$ is $r$-primitive.
Infinite Families

Theorem (Hartke, Stolee, 2012) Let $t \geq 1$,

$$n = 4t^2 + 1, \quad \text{and} \quad r = 2t^2 - t + 1.$$ 

The Cayley complement $\overline{C}(\mathbb{Z}_n, \{1, 2t\})$ is $r$-primitive.

Proof uses counting method.

Conjecture (Hartke, Stolee, 2012) Let $t \geq 1$,

$$n = 9t^2 - 3t + 1 \quad \text{and} \quad r = 3t^2 - 2t + 1.$$ 

The Cayley complement $\overline{C}(\mathbb{Z}_n, \{1, 3t - 1, 3t\})$ is $r$-primitive.
Infinite Families

Theorem (Hartke, Stolee, 2012) Let $t \geq 1$,

$$n = 4t^2 + 1, \quad \text{and} \quad r = 2t^2 - t + 1.$$

The Cayley complement $\overline{C}(\mathbb{Z}_n, \{1, 2t\})$ is $r$-primitive.

Proof uses **counting** method.

Theorem (Hartke, Stolee, 2012) Let $t \geq 1$,

$$n = 9t^2 - 3t + 1 \quad \text{and} \quad r = 3t^2 - 2t + 1.$$

The Cayley complement $\overline{C}(\mathbb{Z}_n, \{1, 3t - 1, 3t\})$ is $r$-primitive.

Proof uses **discharging** method.
Complexity Results in This Thesis

1. ReachFewL = ReachUL.  
   (with Garvin, Tewari, Vinodchandran)  
   Chapter 13

2. Reachability in surface-embedded acyclic digraphs.  
   (with Vinodchandran)  
   Chapter 14
A language is in $L$ if there is a **deterministic** Turing machine that decides the language using at most $O(\log(n))$ work cells.
Space-Bounded Complexity

A language is in $L$ if there is a deterministic Turing machine that decides the language using at most $O(\log(n))$ work cells.

A language is in $NL$ if there is a non-deterministic Turing machine that decides the language using at most $O(\log(n))$ work cells.

$L \subseteq NL$
Configuration Graphs

If $M$ is an $O(\log(n))$-space non-deterministic Turing machine and $x \in \{0, 1\}^*$, the configuration graph $G_{M,x}$ has vertices representing configurations: assignments of state, work cell contents, and tape head positions. An edge $C \rightarrow C'$ exists if there is a transition function of $M$ whose operation on $C$ results in $C'$. $M$ accepts $x$ if and only if there is a path from $C_{\text{init}}$ to $C_{\text{accept}}$ in $G_{M,x}$. The configuration graph $G_{M,x}$ has poly-size and can be written using log-space.
Configuration Graphs

If $M$ is an $O(\log(n))$-space non-deterministic Turing machine and $x \in \{0, 1\}^*$, the configuration graph $G_{M,x}$ has

1. Vertices are configurations: assignments of state, work cell contents, and tape head positions.

   (Requires $O(\log n)$ bits to describe.)
Configuration Graphs

If $M$ is an $O(\log(n))$-space non-deterministic Turing machine and $x \in \{0, 1\}^*$, the **configuration graph** $G_{M,x}$ has

1. Vertices are **configurations**: assignments of state, work cell contents, and tape head positions.  
   
   (Requires $O(\log n)$ bits to describe.)

2. An edge $C \rightarrow C'$ exists if there is a transition function of $M$ whose operation on $C$ results in $C'$. 

Derrick Stolee (UNL)
Configuration Graphs

If $M$ is an $O(\log(n))$-space non-deterministic Turing machine and $x \in \{0, 1\}^*$, the configuration graph $G_{M,x}$ has

1. Vertices are configurations: assignments of state, work cell contents, and tape head positions. (Requires $O(\log n)$ bits to describe.)

2. An edge $C \rightarrow C'$ exists if there is a transition function of $M$ whose operation on $C$ results in $C'$.

$M$ accepts $x$ if and only if there is a path from $C_{\text{init}}$ to $C_{\text{accept}}$ in $G_{M,x}$. 
Configuration Graphs

If $M$ is an $O(\log(n))$-space non-deterministic Turing machine and $x \in \{0, 1\}^*$, the **configuration graph** $G_{M,x}$ has

1. Vertices are **configurations**: assignments of state, work cell contents, and tape head positions.  
   (Requires $O(\log n)$ bits to describe.)

2. An edge $C \rightarrow C'$ exists if there is a transition function of $M$ whose operation on $C$ results in $C'$.

$M$ accepts $x$ if and only if there is a path from $C_{\text{init}}$ to $C_{\text{accept}}$ in $G_{M,x}$.

$G_{M,x}$ has poly-size and can be written using log-space.
Meta-Theory of Space-Bounded Complexity

Every space-bounded complexity problem can be reduced to some form of the \textit{reachability problem} in digraphs.
Meta-Theory of Space-Bounded Complexity

Every space-bounded complexity problem can be reduced to some form of the reachability problem in digraphs.

Reach = \{ \langle G, s, t \rangle : G \text{ is a directed graph with a path from } s \text{ to } t \}\n
L \subseteq NL \subseteq P
Complexity Results in This Thesis

1. \textit{ReachFewL} = \textit{ReachUL}.
   (with Garvin, Tewari, Vinodchandran)  
   Chapter 13

2. Reachability in surface-embedded acyclic digraphs.
   (with Vinodchandran)  
   Chapter 14
Complexity Results in This Thesis

1. ReachFewL = ReachUL.
   (with Garvin, Tewari, Vinodchandran)  
   Chapter 13

2. Reachability in surface-embedded acyclic digraphs.
   (with Vinodchandran)  
   Chapter 14
Log-space Classes and Reachability

\[ L \]
Deterministic

Complete:
Undirected Reach
(Reingold 08)
Log-space Classes and Reachability

**Deterministic**: Undirected Reach (Reingold 08)

**Nondeterministic**: Directed Reach

**Complete**: Undirected Reach (Reingold 08)

**Complete**: Directed Reach
Log-space Classes and Reachability

**L**
- Deterministic
- Complete: Undirected Reach (Reingold 08)

**UL**
- Unambiguous
- Contains: Dir. Planar Reach (Bourke, Tewari, Vinodchandran 09)

**NL**
- Nondeterministic
- Complete: Directed Reach
Other Perspectives

\[
\begin{align*}
\text{L} & \quad \uparrow \\
\text{UL} & \quad \uparrow \\
\text{NL} & \quad \uparrow \\
\text{TISP} & \left[ \text{poly}(n), n^2 \sqrt{\log n} \right] \\
\text{SPACE} & \left[ \log_2 n \right] \\
\text{SPACE} & \left[ \log_2 n - \varepsilon n \right] \\
\text{UL} & \quad \uparrow \\
\text{NL} & \quad \uparrow \\
\text{L} & \quad \uparrow
\end{align*}
\]
Other Perspectives

NL

UL

BTV '09

PlanarReach

?}

L
Other Perspectives

SPACE[$\log^2 n$]

Savitch, '70

NL

UL

BTV '09

PlanarReach

L
Other Perspectives

\[
\text{SPACE}[\log^2 n] \quad \text{Savitch, '70} \quad \text{NL}
\]

\[
\text{SPACE}[\log^{2-\varepsilon} n] \quad \text{UL} \quad \text{BTV '09}
\]

\[
\text{PlanarReach} \quad \text{L}
\]
Other Perspectives

\[
\text{SPACE} \left[ \log^2 n \right] \quad \text{SPACE} \left[ \log^{2-\varepsilon} n \right] \quad \text{TISP} \left[ \text{poly}(n), \frac{n}{2^{\sqrt{\log n}}} \right]
\]

Savitch, '70 \quad \text{BBRS, '92}

PlanarReach \quad \text{BTV '09} \quad \text{L}

Derrick Stolee (UNL)  Computational Combinatorics  58 / 68
Other Perspectives

SPACE[$\log^2 n$] \rightarrow \text{Savitch, '70} \rightarrow \text{BBRS, '92} \rightarrow \text{TISP} \left[ \text{poly}(n), \frac{n}{2^{\sqrt{\log n}}} \right]

SPACE[$\log^{2-\varepsilon} n$] \rightarrow \text{Savitch, '70} \rightarrow \text{BBRS, '92} \rightarrow \text{TISP} \left[ \text{poly}(n), n^{1-\varepsilon} \right]

\text{UL} \rightarrow \text{BTV '09} \rightarrow \text{PlanarReach} \rightarrow \text{L}
Planar and Acyclic Restrictions

Reach for **acyclic** digraphs is complete for NL.
Planar and Acyclic Restrictions

1. Reach for *acyclic* digraphs is complete for NL.

2. Reach for *planar* digraphs is in UL, but we believe UL = NL.
Planar and Acyclic Restrictions

1. Reach for **acyclic** digraphs is complete for NL.
2. Reach for **planar** digraphs is in UL, but we believe UL = NL.
3. What if we combine **acyclic** and **planar**?
Planar and Acyclic Restrictions

1. Reach for **acyclic** digraphs is complete for NL.

2. Reach for **planar** digraphs is in UL, but we believe UL = NL.

3. What if we combine **acyclic** and **planar**?

We also bound number of

[Diagram showing sources and sinks]
Planar + Acyclic Reachability in Log-Space

1. Series-parallel graphs
   (Jakoby, Liśkiewicz, Reischuk, Tantau, ’06/’07)
Planar + Acyclic Reachability in Log-Space

1. Series-parallel graphs  
   (Jakoby, Liśkiewicz, Reischuk, Tantau, ’06/’07)

2. Single-source Single-Sink Planar DAGs  
   (Allender, Barrington, Chakraborty, Datta, Roy, ’09)
Planar + Acyclic Reachability in Log-Space

1. Series-parallel graphs
   (Jakoby, Liśkiewicz, Reischuk, Tantau, ’06/’07)

2. Single-source Single-Sink Planar DAGs
   (Allender, Barrington, Chakraborty, Datta, Roy, ’09)

3. Single-source Multiple-Sink Planar DAGs
   (Allender, Barrington, Chakraborty, Datta, Roy, ’09)
Planar + Acyclic Reachability in Log-Space

1. Series-parallel graphs
   (Jakoby, Liśkiewicz, Reischuk, Tantau, ’06/’07)

2. Single-source Single-Sink Planar DAGs
   (Allender, Barrington, Chakraborty, Datta, Roy, ’09)

3. Single-source Multiple-Sink Planar DAGs
   (Allender, Barrington, Chakraborty, Datta, Roy, ’09)

4. Log-source Multiple-Sink Planar DAGs
   (Stolee, Bourke, Vinodchandran, ’10)
Surface-embedded graphs

We also extend to graphs embedded in *surfaces of low genus*.
Surface-embedded graphs

We also extend to graphs embedded in *surfaces of low genus*.

Let $G(m, g)$ denote the *acyclic* digraphs with *$m$ sources* and embedded in a *genus $g$ surface*. 
Reduction with Compression

**Theorem (Stolee, Vinodchandran, ’12)** Given a graph $G \in \mathcal{G}(m, g)$ and $s, t \in V(G)$, we can compute in log-space a graph $G'$ with vertices $s', t'$ so that

1. There is a path from $s$ to $t$ in $G$ if and only if there is a path from $s'$ to $t'$ in $G'$.
2. $G'$ has $O(m + g)$ vertices.
Topological Equivalence
Topological Equivalence
Topological Equivalence
Topological Equivalence
Topological Equivalence
Topological Equivalence
Our Results (Stolee, Vinodchandran, ’12)

**Theorem (Sub-Savitch)** Reachability for graphs of order $n$ in $G(m, g)$ is in SPACE[$\log n + \log^2(m + g)$].
Our Results

(Stolee, Vinodchandran, ’12)

**Theorem (Sub-Savitch)** Reachability for graphs of order $n$ in $G(m, g)$ is in $\text{SPACE}[\log n + \log^2 (m + g)]$. 

**Theorem (Log-Space)** If $m = g = 2^{\sqrt{\log n}}$, reach for $G(m, g)$ is in $L$. 

**Our Results**  
(Stolee, Vinodchandran, ’12)

**Theorem (Sub-Savitch)** Reachability for graphs of order $n$ in $G(m, g)$ is in $\text{SPACE}[^{\log n + \log^2 (m + g)}]$. 

**Theorem (Log-Space)** If $m = g = 2^{\sqrt{\log n}}$, reach for $G(m, g)$ is in $L$. 

**Theorem (Time-Space)** Reachability for graphs of order $n$ in $G(m, g)$ is in $\text{TISP}[^{\text{poly}(n), \log n + m + g}]$. 
Computational Combinatorics

Computational Complexity
Derrick Stolee

University of Nebraska–Lincoln
s-dstoleel@math.unl.edu
http://www.math.unl.edu/~s-dstoleel/

March 13, 2012
Dissertation Defense

Supported by NSF grants DMS-0354008, DMS-0914815, and CCF-0916525,
and a University of Nebraska Presidential Fellowship.