Covering Arrays and Related Problems in Extremal Combinatorics

Karen Meagher
Joint work with Lucia Moura, University of Ottawa
and Brett Stevens, Carleton University
kmeagher@site.uottawa.ca

University of Ottawa
A covering array $CA(n, k, g)$ is an $k \times n$ array with:

- $k$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- between any two rows all pairs from $\mathbb{Z}_g$ occur (qualitatively independent)
A covering array $CA(n, k, g)$ is an $k \times n$ array with:

- $k$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- between any two rows all pairs from $\mathbb{Z}_g$ occur (qualitatively independent)

A minimal $CA(5, 4, 2)$ so $CAN(4, 2) = 5$
What is a Covering Array?

A covering array $CA(n, k, g)$ is an $k \times n$ array with:

- $k$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- between any two rows all pairs from $\mathbb{Z}_g$ occur (qualitatively independent)

A minimal $CA(5, 4, 2)$ so $CAN(4, 2) = 5$
What is a Covering Array?

A covering array $CA(n, k, g)$ is an $k \times n$ array with:

- $k$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- between any two rows all pairs from $\mathbb{Z}_g$ occur (qualitatively independent)

$$
\begin{array}{cccccc}
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

A minimal $CA(5, 4, 2)$ so $CAN(4, 2) = 5$
What is a Covering Array?

A covering array $CA(n, k, g)$ is an $k \times n$ array with:

- $k$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- between any two rows all pairs from $\mathbb{Z}_g$ occur (qualitatively independent)

A minimal $CA(5, 4, 2)$ so $CAN(4, 2) = 5$
What is a Covering Array?

A covering array $CA(n, k, g)$ is an $k \times n$ array with:

- $k$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_{g}$ ($g$ is the alphabet)
- between any two rows all pairs from $\mathbb{Z}_{g}$ occur
  (qualitatively independent)

\[
\begin{array}{cccccc}
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 \\
\end{array}
\]

A minimal $CA(5, 4, 2)$ so $CAN(4, 2) = 5$
Testing Applications

The 5 parameters for a house party:
Testing Applications

The 5 parameters for a house party:

- Drinks - Beer, Wine or Gin
Testing Applications

The 5 parameters for a house party:

- Drinks - Beer, Wine or Gin
- Snacks - Veggies, Olives or Tofu
Testing Applications

The 5 parameters for a house party:

- Drinks - Beer, Wine or Gin
- Snacks - Veggies, Olives or Tofu
- Attire - Casual, Costume or Formal
The 5 parameters for a house party:

- Drinks - Beer, Wine or Gin
- Snacks - Veggies, Olives or Tofu
- Attire - Casual, Costume or Formal
- Guests - Everybody, Math Kids or Girls Only
Testing Applications

The 5 parameters for a house party:

- Drinks - Beer, Wine or Gin
- Snacks - Veggies, Olives or Tofu
- Attire - Casual, Costume or Formal
- Guests - Everybody, Math Kids or Girls Only
- Time - 7:30, 10:00 or Whenever
Testing Applications

The 5 parameters for a house party:

- Drinks - Beer, Wine or Gin
- Snacks - Veggies, Olives or Tofu
- Attire - Casual, Costume or Formal
- Guests - Everybody, Math Kids or Girls Only
- Time - 7:30, 10:00 or Whenever

To do all combinations we need to have

\[ 3 \times 3 \times 3 \times 3 \times 3 = 243 \text{ parties} \]
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Formal</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Formal</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Formal</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Formal</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Formal</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
# Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Formal</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Formal</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Costume</td>
<td>Math Kids</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Formal</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Costume</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Casual</td>
<td>Math Kids</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Formal</td>
<td>Math Kids</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td><strong>Gin</strong></td>
<td>Veggies</td>
<td>Formal</td>
<td><strong>Math Kids</strong></td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Costume</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Formal</td>
<td>Math Friends</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Costume</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Casual</td>
<td>Math Friends</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Formal</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Costume</td>
<td>Math Friends</td>
<td>Whenever</td>
</tr>
</tbody>
</table>
## Testing the Perfect Party

<table>
<thead>
<tr>
<th>Drinks</th>
<th>Snacks</th>
<th>Attire</th>
<th>Guests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>Veggies</td>
<td>Casual</td>
<td>Everybody</td>
<td>Whenever</td>
</tr>
<tr>
<td>Beer</td>
<td>Olives</td>
<td>Formal</td>
<td>Math Friends</td>
<td>7:30</td>
</tr>
<tr>
<td>Beer</td>
<td>Tofu</td>
<td>Costume</td>
<td>Girls Only</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Veggies</td>
<td>Casual</td>
<td>Math Friends</td>
<td>10:00</td>
</tr>
<tr>
<td>Wine</td>
<td>Olives</td>
<td>Formal</td>
<td>Girls Only</td>
<td>Whenever</td>
</tr>
<tr>
<td>Wine</td>
<td>Tofu</td>
<td>Costume</td>
<td>Everybody</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Veggies</td>
<td>Casual</td>
<td>Girls Only</td>
<td>7:30</td>
</tr>
<tr>
<td>Gin</td>
<td>Olives</td>
<td>Formal</td>
<td>Everybody</td>
<td>10:00</td>
</tr>
<tr>
<td>Gin</td>
<td>Tofu</td>
<td>Costume</td>
<td>Math Friends</td>
<td>Whenever</td>
</tr>
</tbody>
</table>
A covering array on a graph $G$ denoted $CA(n, G, g)$, has:

- $k = |V(G)|$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- rows for adjacent vertices have all pairs from $\mathbb{Z}_g$
A covering array on a graph $G$ denoted $CA(n, G, g)$, has:

- $k = |V(G)|$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- rows for adjacent vertices have all pairs from $\mathbb{Z}_g$
A covering array on a graph $G$ denoted $CA(n, G, g)$, has:

- $k = |V(G)|$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- rows for adjacent vertices have all pairs from $\mathbb{Z}_g$
A covering array on a graph $G$ denoted $CA(n, G, g)$, has:

- $k = |V(G)|$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- rows for adjacent vertices have all pairs from $\mathbb{Z}_g$
A covering array on a graph $G$ denoted $CA(n, G, g)$, has:

- $k = |V(G)|$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- rows for adjacent vertices have all pairs from $\mathbb{Z}_g$

\[
\begin{array}{cccccc}
1 & 0 & 0 & 1 & 1 & 1 \\
2 & 0 & 1 & 0 & 1 & 1 \\
3 & 0 & 1 & 1 & 0 & 1 \\
4 & 0 & 0 & 1 & 1 & 1 \\
5 & 0 & 1 & 0 & 1 & 1 \\
6 & 0 & 1 & 1 & 0 & 1 \\
7 & 0 & 1 & 1 & 1 & 0 \\
\end{array}
\]
A covering array on a graph $G$ denoted $\text{CA}(n, G, g)$, has:

- $k = |V(G)|$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- rows for adjacent vertices have all pairs from $\mathbb{Z}_g$
A covering array on a graph $G$ denoted $CA(n, G, g)$, has:

- $k = |V(G)|$ rows of length $n$ ($n$ is the size)
- entries from $\mathbb{Z}_g$ ($g$ is the alphabet)
- rows for adjacent vertices have all pairs from $\mathbb{Z}_g$
Graph Homomorphisms

A homomorphism, \( f : G \rightarrow H \) is

* an edge preserving map between two graphs
* if vertices \( u, v \in G \) are adjacent then vertices \( f(u), f(v) \in H \) are also adjacent
A homomorphism, $f : G \rightarrow H$ is
- an edge preserving map between two graphs
- if vertices $u, v \in G$ are adjacent then vertices $f(u), f(v) \in H$ are also adjacent
Normal covering arrays correspond to covering arrays on complete graphs

\[ \text{CAN}(K_k, g) = \text{CAN}(k, g). \]
Motivation and Direction

- Normal covering arrays correspond to covering arrays on complete graphs

\[ CAN(K_k, g) = CAN(k, g). \]

- There is a bound from max clique size,
Motivation and Direction

- Normal covering arrays correspond to covering arrays on complete graphs
  \[ CAN(K_k, g) = CAN(k, g) \].
- There is a bound from max clique size,
  \[ CAN(\omega(G), g) \leq CAN(G, g) \].
Motivation and Direction

- Normal covering arrays correspond to covering arrays on complete graphs
  \[ \text{CAN}(K_k, g) = \text{CAN}(k, g). \]
- There is a bound from max clique size,
  \[ \text{CAN}(\omega(G), g) \leq \text{CAN}(G, g) \leq \text{CAN}(\chi(G), g). \]
- and a bound from the chromatic number.
Define graph \( QI(n, g) \) by

- Vertices are all vectors which could go into a row of a covering array of size \( n \) on \( \mathbb{Z}_g \).
- Vertices are adjacent iff the vectors have all pairs from \( \mathbb{Z}_g \).
Define graph $QI(n, g)$ by

- Vertices are all vectors which could go into a row of a covering array of size $n$ on $\mathbb{Z}_g$.
- Vertices are adjacent iff the vectors have all pairs from $\mathbb{Z}_g$. 

\[
\begin{array}{cccc}
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 
\end{array}
\]
Define graph $QI(n, g)$ by

- Vertices are all vectors which could go into a row of a covering array of size $n$ on $\mathbb{Z}_g$.
- Vertices are adjacent iff the vectors have all pairs from $\mathbb{Z}_g$.
Define graph \( QI(n, g) \) by

- Vertices are all vectors which could go into a row of a covering array of size \( n \) on \( \mathbb{Z}_g \).
- Vertices are adjacent iff the vectors have all pairs from \( \mathbb{Z}_g \).

The graph \( QI(4, 2) \).
Why is $QI(n, g)$ so Good?

This family determines up to homomorphism the covering number for a graph.

**Theorem (M. and Stevens, 2002)**

A $CA(n, G, g)$ exists if and only if there is a homomorphism

$$G \rightarrow QI(n, g).$$
Parallel theorems!

Theorem (M. and Stevens, 2002)

\[ CAN(G, g) = \min\{ n ; G \rightarrow QI(n, g) \} \]
Theorem (M. and Stevens, 2002)

$$CAN(G, g) = \min\{n : G \rightarrow QI(n, g)\}$$

$$\chi(G) = \min\{n : G \rightarrow K_n\}$$
Parallel theorems!

**Theorem (M. and Stevens, 2002)**

\[
CAN(G, g) = \min \{n ; G \rightarrow QI(n, g)\}
\]

\[
\chi(G) = \min \{n ; G \rightarrow K_n\}
\]

\[
\chi^*(G) = \min \{v/r ; G \rightarrow K_{v,r}\}
\]
Parallel theorems!

Theorem (M. and Stevens, 2002)

\[ CAN(G, g) = \min\{n ; G \rightarrow QI(n, g)\} \]

\[ \chi(G) = \min\{n ; G \rightarrow K_n\} \]

\[ \chi^*(G) = \min\{v/r ; G \rightarrow K_{v,r}\} \]

\[ \chi_c(G) = \min\{v/r ; G \rightarrow C(v, r)\} \]
The graph $QI(4, 2)$ is isomorphic to $K_3$. 
The graph $\mathcal{QI}(4, 2)$ is isomorphic to $K_3$.

Determining if a graph has a binary covering array of size 4 is the same as determining if a graph is 3-colourable (NP-complete, Seroussi and Bshouty, 1988).
The Graph $QI(5, 2)$

Interesting facts about this graph:

- $V(QI(5, 2)) = 10$
- $\omega(QI(5, 2)) = 4$
- $\chi(QI(5, 2)) = 5$
The Graph $QI(5, 2)$

Interesting facts about this graph

- $V(QI(5, 2)) = 10$
- $\omega(QI(5, 2)) = 4$
- $\chi(QI(5, 2)) = 5$

$CAN(QI(5, 2), 2) = 5$
The Graph $QI(5, 2)$

Interesting facts about this graph:

- $V(QI(5, 2)) = 10$
- $\omega(QI(5, 2)) = 4$
- $\chi(QI(5, 2)) = 5$

$CAN(QI(5, 2), 2) = 5 < 6 = CAN(5, 2)$. 
Facts for $QI(n, 2)$

Theorem (Kleitman and Spencer 1973, Katona 1973)

$$\omega(QI(n, 2)) = \left(\left\lfloor \frac{n}{2} \right\rfloor - 1 \right).$$
## Facts for $QI(n, 2)$

<table>
<thead>
<tr>
<th>Theorem (Kleitman and Spencer 1973, Katona 1973)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega(QI(n, 2)) = \left( \left\lfloor \frac{n}{2} \right\rfloor - 1 \right)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem (M. and Stevens 2002)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi(QI(n, 2)) = \left\lfloor \frac{1}{2} \left( \left\lfloor \frac{n}{2} \right\rfloor \right) \right\rfloor$</td>
</tr>
</tbody>
</table>
Facts for $QI(n, 2)$

Theorem (Kleitman and Spencer 1973, Katona 1973)

$$\omega(QI(n, 2)) = \left( \frac{n - 1}{\lfloor \frac{n}{2} \rfloor - 1} \right).$$

Theorem (M. and Stevens 2002)

$$\chi(QI(n, 2)) = \left\lceil \frac{1}{2} \left( \binom{n}{\lfloor \frac{n}{2} \rfloor} \right) \right\rceil.$$ 

Theorem (M. and Stevens 2002)

For $n$ even

$$CAN(QI(n, 2), 2) = CAN(\chi(QI(n, 2)), 2),$$

and for $n$ odd

$$CAN(QI(n, 2), 2) = CAN(\chi(QI(n, 2)), 2) - 1.$$
The vertices of $QI(n, 2)$ are characteristic vectors for subsets. For example:
Connection to Set Systems

The vertices of $QI(n, 2)$ are characteristic vectors for subsets. For example:

$0 0 1 1 1$ and $0 1 0 1 1$ and $0 0 1 1 0$
Connection to Set Systems

The vertices of $QI(n, 2)$ are characteristic vectors for subsets.
For example:

\[
\begin{align*}
0 & 0 & 1 & 1 & 1 & \quad \text{and} \quad 0 & 1 & 0 & 1 & 1 & \quad \text{and} \quad 0 & 0 & 1 & 1 & 0 \\
1 & 2 & 3 & 4 & 5 & \quad \text{and} \quad 1 & 2 & 3 & 4 & 5 & \quad \text{and} \quad 1 & 2 & 3 & 4 & 5
\end{align*}
\]
The vertices of $QI(n, 2)$ are characteristic vectors for subsets. For example:

$0 \ 0 \ 1 \ 1 \ 1$ and $0 \ 1 \ 0 \ 1 \ 1$ and $0 \ 0 \ 1 \ 1 \ 0$

$1 \ 2 \ 3 \ 4 \ 5$ and $1 \ 2 \ 3 \ 4 \ 5$ and $1 \ 2 \ 3 \ 4 \ 5$

$\{3, 4, 5\}$ and $\{2, \ 4, 5\}$ and $\{3, 4\}$
Connection to Set Systems

The vertices of $QI(n, 2)$ are characteristic vectors for subsets. For example:

\begin{align*}
0 & \quad 0 \quad 1 \quad 1 \quad 1 \\
1 \quad 2 \quad 3 \quad 4 \quad 5 & \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \\
\{1, 2\} & \quad \{1, 3\} \quad \text{and} \quad \{3, 4\}
\end{align*}
Connection to Set Systems

The vertices of $QI(n, 2)$ are characteristic vectors for subsets. For example:

- $0\ 0\ 1\ 1\ 1$ and $0\ 1\ 0\ 1\ 1$ and $0\ 0\ 1\ 1\ 0$ for $n=5$
- $\{1,\ 2\}$ and $\{1,\ 3\}$ and $\{3,\ 4\}$

To be qualitatively independent sets are

- intersecting
- not contain each other
Mapping to the Core
Mapping to the Core
Mapping to the Core

- consider vectors with less than half 1’s

1100 1010 1001 0110 0101 0011

1000 0100

0010 0001

0000
Mapping to the Core

consider vectors with less than half 1’s

\begin{align*}
1100 & \quad 1010 & \quad 1001 & \quad 0110 & \quad 0101 & \quad 0011 \\
1000 & \quad 0100 & \quad 0010 & \quad 0001 & \\
& \quad & \quad & \quad & \quad & \quad & 0000
\end{align*}
consider vectors with less than half 1’s

1100  1010  1001  0110  0101  0011

1000  0100  0010  0001

0000
Mapping to the Core

- Consider vectors with less than half 1’s

1100  1010  1001  0110  0101  0011

- Use chains to map to \( \lceil n/2 \rceil \) level
Mapping to the Core

- consider vectors with less than half 1’s

```
1100  1010  1001  0110  0101  0011
1000  0100  0010  0001
0000
```

- use chains to map to \( \lfloor n/2 \rfloor \) level
- map preserves qualitative independence
Mapping to the Core

- consider vectors with less than half 1’s

\[
\begin{align*}
1100 & \quad 1010 & \quad 1001 & \quad 0110 & \quad 0101 & \quad 0011 \\
1000 & \quad 0100 & \quad 0010 & \quad 0001 & \quad 0000
\end{align*}
\]

- use chains to map to \( \lfloor n/2 \rfloor \) level
- map preserves qualitative independence
- graph with vertices from \( \lfloor n/2 \rfloor \) level is core
The Core of $QI(n, 2)$

Theorem (M. and Stevens 2002)
For $n$ even the core of $QI(n, 2)$ is $K_{\frac{1}{2}(\frac{n}{2})}$. For $n$ odd the core is the subgraph induced by vertices with $\frac{n-1}{2}$ 1’s.
The Core of $QI(n, 2)$

Theorem (M. and Stevens 2002)
For $n$ even the core of $QI(n, 2)$ is $K_{\frac{1}{2}}(\frac{n}{2})$.
For $n$ odd the core is the subgraph induced by vertices with $\frac{n-1}{2}$ 1’s.

We can just use the vertices of $QI(n, 2)$ which correspond to $\lfloor n/2 \rfloor$ sets of $n$
A maximum system of intersecting \( \lceil n/2 \rceil \)-sets of \( n \) is a maximum clique in \( QI(n, 2) \).
A maximum system of intersecting $\lceil n/2 \rceil$-sets of $n$ is a maximum clique in $QI(n, 2)$. 
The Erdős-Ko-Rado theorem

**Intersecting Set System** A $k$-set system with all sets pairwise intersecting.

<table>
<thead>
<tr>
<th></th>
<th>123</th>
<th>124</th>
<th>125</th>
<th>126</th>
<th>134</th>
</tr>
</thead>
<tbody>
<tr>
<td>135</td>
<td>136</td>
<td>145</td>
<td>146</td>
<td>234</td>
<td></td>
</tr>
</tbody>
</table>
The Erdős-Ko-Rado theorem

Trivial Intersecting Set System All sets have a common element

\begin{align*}
123 & \quad 124 & \quad 125 & \quad 126 & \quad 134 \\
135 & \quad 136 & \quad 145 & \quad 146 & \quad 156
\end{align*}

For $n = 6$ and $k = 3$ size is $\binom{5}{2} = 10$
The Erdős-Ko-Rado theorem

Trivial Intersecting Set System All sets have a common element

\[
\begin{array}{cccccc}
1 & 2 & 3 \\
1 & 2 & 4 \\
1 & 2 & 5 \\
1 & 2 & 6 \\
1 & 3 & 4 \\
1 & 3 & 5 \\
1 & 3 & 6 \\
1 & 4 & 5 \\
1 & 4 & 6 \\
1 & 5 & 6 \\
\end{array}
\]

For \( n = 6 \) and \( k = 3 \) size is \( \binom{5}{2} = 10 \)

**Theorem (Erdős-Ko-Rado 1961)** For \( n \geq 2k \), the maximal intersecting \( k \)-set system is a trivial system, if \( n > 2k \), this maximum is unique.
This solves $CAN(k, 2)$

**Corollary** \[ \omega(QI(n, 2)) = \left( \left\lfloor \frac{n-1}{2} \right\rfloor - 1 \right). \]
This solves $CAN(k, 2)$

Corollary \[ \omega(QI(n, 2)) = \left(\frac{n-1}{\lfloor \frac{n}{2} \rfloor} - 1\right). \]

A clique in $QI(n, g)$ is a set of qualitatively independent vertices.

A $CA(n, k, g)$ exists iff $\omega(QI(n, g)) \geq k$. 
This solves $CAN(k, 2)$

Corollary: $\omega(QI(n, 2)) = \left(\frac{n-1}{\lceil n/2 \rceil - 1}\right)$.

A clique in $QI(n, g)$ is a set of qualitatively independent vertices.

A $CA(n, k, g)$ exists iff $\omega(QI(n, g)) \geq k$

Corollary: $CAN(k, 2) = \min\{n \mid k \leq \left(\frac{n-1}{\lceil n/2 \rceil - 1}\right)\}$. 

Covering Arrays and Related Problems in Extremal Combinatorics – p.19/28
This almost solves $CAN(G, g)$

For any graph $G$

$CAN(\chi(G), 2) - 1 \leq CAN(G, 2) \leq CAN(\chi(G), 2)$. 
This almost solves $CAN(G, g)$

For any graph $G$

$CAN(\chi(G), 2) - 1 \leq CAN(G, 2) \leq CAN(\chi(G), 2)$.

We have necessary conditions on $G$ for

$CAN(G, 2) < CAN(\chi(G), 2)$.

These are:

- $CAN(\chi(G), 2) = c$ must be even, and

- $\left(\frac{c-2}{2}\right) < \chi(G) \leq \left\lceil \frac{1}{2} \left(\frac{c-1}{2}\right) \right\rceil$. 
In the general case, the vertices of $QI(n, g)$ are like $g$-ary partitions.

For example:
Beyond the Binary

In the general case, the vertices of $QI(n, g)$ are like $g$-ary partitions.

For example:

0 0 0 1 1 1 2 2 2

0 1 2 0 1 2 0 1 2
In the general case, the vertices of $QI(n, g)$ are like $g$-ary partitions.

For example:

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}
\]

\[
\begin{array}{cccccccc}
0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}
\]
In the general case, the vertices of $QI(n, g)$ are like $g$-ary partitions.

For example:

<table>
<thead>
<tr>
<th>0 0 0 1 1 1 2 2 2 2</th>
<th>0 1 2 0 1 2 0 1 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8 9</td>
<td>1 2 3 4 5 6 7 8 9</td>
</tr>
<tr>
<td>1 2 3</td>
<td>4 5 6</td>
</tr>
</tbody>
</table>
Big Question

* Can we get similar results on the systems of intersecting partitions

What are intersecting partitions?

What would intersecting partitions correspond to in $QI(n;g)$?
Big Question

★ Can we get similar results on the systems of intersecting partitions
★ Can these be used for covering arrays with larger alphabets?
Big Question

- Can we get similar results on the systems of intersecting partitions?
- Can these be used for covering arrays with larger alphabets?
- What are intersecting partitions?
Can we get similar results on the systems of intersecting partitions?

Can these be used for covering arrays with larger alphabets?

What are intersecting partitions?

What would intersecting partitions correspond to in $QI(n, g)$?
Uniform Partitions

In $QI(g^2, g)$ partitions correspond to adjacent vertices if no pairs in any class are repeated:
Uniform Partitions

In $QI(g^2, g)$ partitions correspond to adjacent vertices if no pairs in any class are repeated:

$$\begin{array}{cccccc}
0 & 0 & 0 & 1 & 1 & 1 \\
1 & 2 & 3 & 4 & 5 & 6 \\
7 & 8 & 9 & & & \\
\end{array}$$

$$\begin{array}{cccccc}
0 & 1 & 2 & 0 & 1 & 2 \\
1 & 4 & 7 & 2 & 5 & 8 \\
3 & 6 & 9 & & & \\
\end{array}$$
Uniform Partitions

In $QI(g^2, g)$ partitions correspond to adjacent vertices if no pairs in any class are repeated:

- Uniform Partition A partition which has all classes the same size.
Partitions

\[ A = \{ A_1, A_2, \ldots, A_g \} \quad B = \{ B_1, B_2, \ldots, B_g \} \]

are \textit{t-partially intersecting} if there exist an \( i \) and a \( j \) so that

\[ |A_i \cap B_j| \geq t. \]
$t$-Partially Intersecting

Partitions

$$A = \{ A_1, A_2, \ldots, A_g \} \quad B = \{ B_1, B_2, \ldots, B_g \}$$

are $t$-partially intersecting if there exist an $i$ and a $j$ so that

$$|A_i \cap B_j| \geq t.$$ 

$t$-partially intersecting partition system A partition system with all partitions pairwise $t$-partially intersecting.
A Trivial System

A *trivial* 2-partially intersecting system

\[
\begin{array}{ccc|ccc|ccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 2 & 3 & 4 & 5 & 7 & 6 & 8 & 9 \\
& & & & & & & & \\
1 & 2 & 9 & 3 & 7 & 8 & 4 & 5 & 6 \\
\end{array}
\]

Conjecture

Any maximum \(t\)-partially intersecting uniform partition system is trivial.

Conjecture

\[
Q_I(g_1;g_2) = g_2^2 g_2 (g_1)! (g_1)! (g_1)!
\]
A Trivial System

A trivial 2-partially intersecting system

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 & 3 \\
\vdots \\
1 & 2 & 9
\end{array}
\ \begin{array}{ccc}
4 & 5 & 6 \\
4 & 5 & 7 \\
3 & 7 & 8 \\
3 & 7 & 8
\end{array}
\ \begin{array}{c}
7 & 8 & 9 \\
6 & 8 & 9 \\
4 & 5 & 6
\end{array}
\]

Size of system is

\[
\binom{gc - t}{c - t} \frac{((g - 1)c)!}{(g - 1)!c!(g - 1)!}.
\]
A trivial 2-partially intersecting system

<table>
<thead>
<tr>
<th>1 2 3</th>
<th>4 5 6</th>
<th>7 8 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3</td>
<td>4 5 7</td>
<td>6 8 9</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 2 9</td>
<td>3 7 8</td>
<td>4 5 6</td>
</tr>
</tbody>
</table>

Size of system is

\[
\binom{gc - t}{c - t} \frac{((g - 1)c)!}{(g - 1)!(c!(g-1))}.
\]

Conjecture Any maximum \( t \)-partially intersecting uniform partition system is trivial.
A Trivial System

A *trivial* 2-partially intersecting system

<table>
<thead>
<tr>
<th>1 2 3</th>
<th>4 5 6</th>
<th>7 8 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3</td>
<td>4 5 7</td>
<td>6 8 9</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 2 9</td>
<td>3 7 8</td>
<td>4 5 6</td>
</tr>
</tbody>
</table>

Size of system is

\[
\binom{gc - t}{c - t} \frac{((g - 1)c)!}{(g - 1)!((c!)^{g-1})}. 
\]

Conjecture

Any maximum *t*-partially intersecting uniform partition system is trivial.

Conjecture

\[ \alpha(QI(g^2, g)) = \left( \frac{g^2-2}{g-2} \right) \frac{((g-1)g)!}{(g-1)!(g!)^{(g-1)}}. \]
Intersecting Partitions

Partitions

\[ A = \{ A_1, A_2, \ldots, A_g \} \quad B = \{ B_1, B_2, \ldots, B_g \} \]

are *intersecting* if there exist an \( i \) and a \( j \)

\[ A_i = B_j. \]
Intersecting Partitions

Partitions

\[ A = \{ A_1, A_2, \ldots, A_g \} \quad B = \{ B_1, B_2, \ldots, B_g \} \]

are *intersecting* if there exist an \( i \) and a \( j \)

\[ A_i = B_j. \]

**Intersecting Partition System** A partition system with all partitions pairwise intersecting.
A Trivial System

A trivial intersecting system

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Size of system is $(g_1^c)! (g_1)! (c)!$.

Conjecture

Any maximum intersecting uniform partition system is trivial.
A *trivial* intersecting system

| 1 2 3 | 4 5 6 | 7 8 9 | Size of system is \[
\frac{((g - 1)c)!}{(g - 1)! (c!)^{g-1}}.\]  
| 1 2 3 | 4 5 7 | 6 8 9 |
| *...* | *...* | *...* |
| 1 2 3 | 4 8 9 | 5 6 7 |
A Trivial System

A trivial intersecting system

\[
\begin{array}{ccc|ccc}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & 3 & 4 & 5 & 7 \\
1 & 2 & 3 & 4 & 8 & 9 \\
\vdots & & \vdots & & \vdots & \\
1 & 2 & 3 & 4 & 8 & 9 \\
1 & 2 & 3 & 5 & 6 & 7 \\
\end{array}
\]

Size of system is

\[
\frac{((g-1)c)!}{(g-1)!(c!)^{(g-1)}}.
\]

**Conjecture** Any maximum intersecting uniform partition system is trivial.
A Trivial System

A *trivial* intersecting system

<table>
<thead>
<tr>
<th>1 2 3</th>
<th>4 5 6</th>
<th>7 8 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3</td>
<td>4 5 7</td>
<td>6 8 9</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 2 3</td>
<td>4 8 9</td>
<td>5 6 7</td>
</tr>
</tbody>
</table>

Size of system is

\[
\frac{((g - 1)c)!}{(g - 1)! (c!)^{(g - 1)}}.
\]

**Conjecture** Any maximum intersecting uniform partition system is trivial.

**Theorem (M. and Moura 2004)** Let \( g \geq 1, c \neq 2 \) and \( n = gc \). A largest intersecting uniform \( g \)-partition system must be trivially intersecting.
Conclusion

- The graphs $QI(n, g)$ will be useful for studying covering arrays.
Conclusion

- The graphs $QI(n, g)$ will be useful for studying covering arrays
- What are $\omega(QI(n, g)), \chi(QI(n, g))$ and $\alpha(QI(n, g))$?
Conclusion

- The graphs $QI(n, g)$ will be useful for studying covering arrays
- What are $\omega(QI(n, g))$, $\chi(QI(n, g))$ and $\alpha(QI(n, g))$?
- What is the core of $QI(n, g)$?
The graphs $QI(n, g)$ will be useful for studying covering arrays.

What are $\omega(QI(n, g)), \chi(QI(n, g))$ and $\alpha(QI(n, g))$?

What is the core of $QI(n, g)$?

Can we find a Sperner type theorem for partitions?
The graphs $QI(n, g)$ will be useful for studying covering arrays.

What are $\omega(QI(n, g))$, $\chi(QI(n, g))$, and $\alpha(QI(n, g))$?

What is the core of $QI(n, g)$?

Can we find a Sperner type theorem for partitions?

Can we find an Erdős-Ko-Rado type theorem for partitions?
Conclusion

- The graphs $QI(n, g)$ will be useful for studying covering arrays
- What are $\omega(QI(n, g)), \chi(QI(n, g))$ and $\alpha(QI(n, g))$?
- What is the core of $QI(n, g)$?
- Can we find a Sperner type theorem for partitions?
- Can we find an Erdős-Ko-Rado type theorem for partitions?
- Is the maximum (partially $t$-)intersecting uniform $g$-partition system a trivial system?
The graphs $QI(n, g)$ will be useful for studying covering arrays.

What are $\omega(QI(n, g))$, $\chi(QI(n, g))$ and $\alpha(QI(n, g))$?

What is the core of $QI(n, g)$?

Can we find a Sperner type theorem for partitions?

Can we find an Erdős-Ko-Rado type theorem for partitions?

Is the maximum (partially $t$-)intersecting uniform $g$-partition system a trivial system?

Can intersecting partitions be used for covering arrays?