Locating arrays and disjoint partitions

Daniel Horsley (Monash University, Australia)

Joint work with Charles Colbourn and Bingli Fan.

Covering arrays

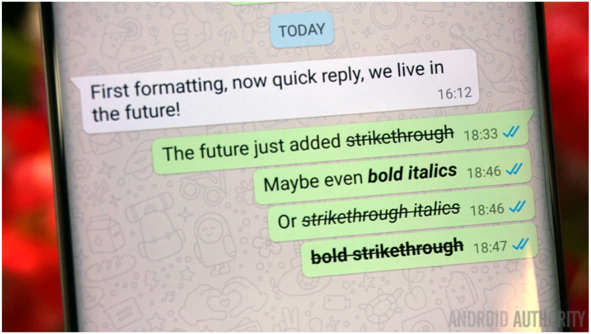
A fault detection problem

A fault detection problem

WhatsApp text formatting rolls out, including bold, italics and strikethrough in messages

by: KRIS CARLON
MARCH 29, 2016

1.8K shares | f 1814 | | g+ 28



A fault detection problem

A fault detection problem

Suppose WhatsApp wants to test its new features:

- B **bold**
- I *italic*
- U underline
- abc ~~strikethrough~~
- A subscript

A fault detection problem

Suppose WhatsApp wants to test its new features:

- B **bold**
- I *italic*
- U underline
- abc ~~strikethrough~~
- A subscript

Testing every possible combination of these would require 32 tests:

B	I	U	abc	A	
0	0	0	0	0	test
0	0	0	0	1	test
0	0	0	1	0	~~test~~
⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	0	***~~test~~***
1	1	1	1	1	***~~test~~***

A fault detection problem

A fault detection problem

Assumption: Faults are caused by interactions of at most two settings.

A fault detection problem

Assumption: Faults are caused by interactions of at most two settings.

Examples:

abc
0

A *1-way interaction*

B	U
1	0

A *2-way interaction*

A fault detection problem

Assumption: Faults are caused by interactions of at most two settings.

Examples:

~~abc~~
0

A *1-way interaction*

B	<u>U</u>
1	0

A *2-way interaction*

How many tests do we need now?

Covering arrays

Covering arrays

We can make do with 6 tests:

Covering arrays

We can make do with 6 tests:

B	*I*	U	~~abc~~	A
0	0	1	0	0
0	1	0	1	1
1	0	0	1	0
1	1	1	0	1
0	1	0	0	0
0	0	1	1	1

A *covering array* with $N = 6$, $k = 5$ and $v = 2$

Covering arrays

We can make do with 6 tests:

B	*I*	U	~~abc~~	A
0	0	1	0	0
0	1	0	1	1
1	0	0	1	0
1	1	1	0	1
0	1	0	0	0
0	0	1	1	1

A *covering array* with $N = 6$, $k = 5$ and $v = 2$

Reveals if there's a faulty 1- or 2-way interaction. (The *strength* is 2.)

Covering arrays

We can make do with 6 tests:

B	I	U	abc	A
0	0	1	0	0
0	1	0	1	1
1	0	0	1	0
1	1	1	0	1
0	1	0	0	0
0	0	1	1	1

A *covering array* with $N = 6$, $k = 5$ and $v = 2$

Reveals if there's a faulty 1- or 2-way interaction. (The *strength* is 2.)

For example, if

B	U
1	0

is faulty then the third test will be failed.

Covering arrays

We can make do with 6 tests:

B	*I*	U	~~abc~~	A
0	0	1	0	0
0	1	0	1	1
1	0	0	1	0
1	1	1	0	1
0	1	0	0	0
0	0	1	1	1

A *covering array* with $N = 6$, $k = 5$ and $v = 2$

Reveals if there's a faulty 1- or 2-way interaction. (The *strength* is 2.)

For example, if

B	U
1	0

is faulty then the third test will be failed.

This approach becomes vital when k is big. For $k = 3000$ and $v = 2$, there exists a strength 2 covering array with 15 rows.

Covering arrays

We can make do with 6 tests:

B	I	U	abc	A
0	0	1	0	0
0	1	0	1	1
1	0	0	1	0
1	1	1	0	1
0	1	0	0	0
0	0	1	1	1

A *covering array* with $N = 6$, $k = 5$ and $v = 2$

Reveals if there's a faulty 1- or 2-way interaction. (The *strength* is 2.)

For example, if

B	U
1	0

is faulty then the third test will be failed.

This approach becomes vital when k is big. For $k = 3000$ and $v = 2$, there exists a strength 2 covering array with 15 rows. And $15 \ll 2^{3000}$.

The covering array problem

The covering array problem

- A piece of software has k parameters; each can take one of v values.
- We know faults are caused only by t-way interactions.
- We can test the software on any assignment of values to parameters and obtain a pass or a fail.
- We wish to prespecify a schedule of N tests after which we will be able to determine if a fault exists (for N as small as possible).

The covering array problem

- A piece of software has k parameters; each can take one of v values.
- We know faults are caused only by t-way interactions.
- We can test the software on any assignment of values to parameters and obtain a pass or a fail.
- We wish to prespecify a schedule of N tests after which we will be able to determine if a fault exists (for N as small as possible).

The covering array problem

- A piece of software has k parameters; each can take one of v values.
- We know faults are caused only by t-way interactions.
- We can test the software on any assignment of values to parameters and obtain a pass or a fail.
- We wish to prespecify a schedule of N tests after which we will be able to determine if a fault exists (for N as small as possible).

We can give a solution to this problem as a strength t covering array.

Covering arrays

Covering arrays

- Covering arrays are used extensively in software testing.

Covering arrays

- Covering arrays are used extensively in software testing.
- They've been well-studied by mathematicians and computer scientists.

Covering arrays

- Covering arrays are used extensively in software testing.
- They've been well-studied by mathematicians and computer scientists.
- For fixed t and v, the minimum number of rows in a strength t covering array with k columns and v symbols is $\Theta(\log k)$.

Covering arrays

- Covering arrays are used extensively in software testing.
- They've been well-studied by mathematicians and computer scientists.
- For fixed t and v, the minimum number of rows in a strength t covering array with k columns and v symbols is $\Theta(\log k)$. The exact value has been determined only for $t = v = 2$.

Covering arrays

- Covering arrays are used extensively in software testing.
- They've been well-studied by mathematicians and computer scientists.
- For fixed t and v, the minimum number of rows in a strength t covering array with k columns and v symbols is $\Theta(\log k)$. The exact value has been determined only for $t = v = 2$.
- The symbols in any column of a covering array can be permuted.

Covering arrays

- Covering arrays are used extensively in software testing.

- They've been well-studied by mathematicians and computer scientists.

- For fixed t and v, the minimum number of rows in a strength t covering array with k columns and v symbols is $\Theta(\log k)$. The exact value has been determined only for $t = v = 2$.

- The symbols in any column of a covering array can be permuted.

- Covering arrays with strength 1 are trivial. For example,

B	*I*	U	~~abc~~	A
0	0	0	0	0
1	1	1	1	1

Locating arrays

Locating faults

Locating faults

Suppose $\dfrac{\boxed{\mathbf{B}} \quad \boxed{\underline{\mathrm{U}}}}{1 \qquad 0}$ is faulty.

Locating faults

Suppose

B	U
1	0

is faulty.

B	I	U	~~abc~~	A
0	0	1	0	0
0	1	0	1	1
1	0	0	1	0
1	1	1	0	1
0	1	0	0	0
0	0	1	1	1

Locating faults

Suppose

B	U
1	0

is faulty.

B	I	U	abc	A	
0	0	1	0	0	pass
0	1	0	1	1	pass
1	0	0	1	0	fail
1	1	1	0	1	pass
0	1	0	0	0	pass
0	0	1	1	1	pass

Locating faults

Suppose

B	U
1	0

is faulty.

B	I	U	abc	A	
0	0	1	0	0	pass
0	1	0	1	1	pass
1	0	0	1	0	fail
1	1	1	0	1	pass
0	1	0	0	0	pass
0	0	1	1	1	pass

We would get the same pass/fail pattern if

I	U
0	0

were faulty.

Locating faults

Suppose (B = 1, U = 0) is faulty.

B	*I*	U	~~abc~~	A	
0	0	1	0	0	pass
0	1	0	1	1	pass
1	0	0	1	0	fail
1	1	1	0	1	pass
0	1	0	0	0	pass
0	0	1	1	1	pass

We would get the same pass/fail pattern if (I = 0, U = 0) were faulty.

A *locating array* is a covering array which also allows faults to be identified.

Locating faults

Suppose $\frac{\mathbf{B}\quad \underline{\text{U}}}{1\quad 0}$ is faulty.

B	*I*	U	~~abc~~	A	
0	0	1	0	0	pass
0	1	0	1	1	pass
1	0	0	1	0	fail
1	1	1	0	1	pass
0	1	0	0	0	pass
0	0	1	1	1	pass

We would get the same pass/fail pattern if $\frac{\textit{I}\quad \underline{\text{U}}}{0\quad 0}$ were faulty.

A *locating array* is a covering array which also allows faults to be identified.

Locating arrays with strength 1 are *not* trivial.

Locating faults

Suppose

B	U
1	0

is faulty.

B	I	U	~~abc~~	A	
0	0	1	0	0	pass
0	1	0	1	1	pass
1	0	0	1	0	fail
1	1	1	0	1	pass
0	1	0	0	0	pass
0	0	1	1	1	pass

We would get the same pass/fail pattern if

I	U
0	0

were faulty.

A *locating array* is a covering array which also allows faults to be identified.

Locating arrays with strength 1 are *not* trivial.
In fact, this talk will be about locating arrays with strength 1.

Our problem

Our problem

- A piece of software has k parameters; each can take one of v values.
- We know the software is faulty on *at most one 1-way interaction*.
- We can test the software on any assignment of values to parameters and obtain a pass or a fail.
- We wish to prespecify a schedule of N tests after which we will be able to *identify the fault* (for N as small as possible).

Our problem

- A piece of software has k parameters; each can take one of v values.
- We know the software is faulty on *at most one 1-way interaction*.
- We can test the software on any assignment of values to parameters and obtain a pass or a fail.
- We wish to prespecify a schedule of N tests after which we will be able to *identify the fault* (for N as small as possible).

Our problem

- A piece of software has k parameters; each can take one of v values.
- We know the software is faulty on *at most one 1-way interaction*.
- We can test the software on any assignment of values to parameters and obtain a pass or a fail.
- We wish to prespecify a schedule of N tests after which we will be able to *identify the fault* (for N as small as possible).

We give a solution to this problem as a $(\overline{1}, 1)$-*locating array*.

Locating arrays

Locating arrays

Not a $(\overline{1}, 1)$-locating array with $N = 6$, $k = 9$ and $v = 3$:

$$\begin{matrix} 1 & 3 & 3 & 3 & 2 & 2 & 1 & 3 & 2 \\ 2 & 1 & 3 & 3 & 3 & 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 3 & 3 & 3 & 1 & 2 & 3 \\ 3 & 2 & 2 & 1 & 3 & 3 & 3 & 1 & 3 \\ 3 & 3 & 2 & 2 & 1 & 3 & 2 & 3 & 1 \\ 3 & 3 & 3 & 2 & 2 & 1 & 3 & 2 & 2 \end{matrix}$$

Locating arrays

Not a $(\overline{1}, 1)$-locating array with $N = 6$, $k = 9$ and $v = 3$:

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	1
2	2	1	3	3	3	1	2	3
3	2	2	1	3	3	3	1	3
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	2

Locating arrays

Not a $(\overline{1}, 1)$-locating array with $N = 6$, $k = 9$ and $v = 3$:

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	1
2	2	1	3	3	3	1	2	3
3	2	2	1	3	3	3	1	3
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	2

A $(\overline{1}, 1)$-locating array with $N = 6$, $k = 9$ and $v = 3$:

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	3
2	2	1	3	3	3	1	2	1
3	2	2	1	3	3	3	1	2
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	3

Locating arrays

Not a $(\overline{1}, 1)$-locating array with $N = 6$, $k = 9$ and $v = 3$:

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	1
2	2	1	3	3	3	1	2	3
3	2	2	1	3	3	3	1	3
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	2

A $(\overline{1}, 1)$-locating array with $N = 6$, $k = 9$ and $v = 3$:

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	3
2	2	1	3	3	3	1	2	1
3	2	2	1	3	3	3	1	2
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	3

Given k and v, we want to minimise N.

Our problem: Given N and v, find the maximum number of columns in a $(\overline{1}, 1)$-locating array with N rows and v symbols.

Our problem: Given N and v, find the maximum number of columns in a $(\overline{1}, 1)$-locating array with N rows and v symbols.

Similar problems have been considered in the combinatorial group testing literature, but not this one exactly.

Disjoint set partitions

Disjoint set partitions

Disjoint set partitions

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	3
2	2	1	3	3	3	1	2	1
3	2	2	1	3	3	3	1	2
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	3

Disjoint set partitions

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	3
2	2	1	3	3	3	1	2	1
3	2	2	1	3	3	3	1	2
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	3

Equivalently:

$\{1\}$	$\{2,3\}$	$\{4,5,6\}$
$\{2\}$	$\{3,4\}$	$\{1,5,6\}$
$\{3\}$	$\{4,5\}$	$\{1,2,6\}$
$\{4\}$	$\{5,6\}$	$\{1,2,3\}$
$\{5\}$	$\{1,6\}$	$\{2,3,4\}$
$\{6\}$	$\{1,2\}$	$\{3,4,5\}$
$\{1,3\}$	$\{2,5\}$	$\{4,6\}$
$\{2,4\}$	$\{3,6\}$	$\{1,5\}$
$\{3,5\}$	$\{1,4\}$	$\{2,6\}$

Disjoint set partitions

1	3	3	3	2	2	1	3	2
2	1	3	3	3	2	2	1	3
2	2	1	3	3	3	1	2	1
3	2	2	1	3	3	3	1	2
3	3	2	2	1	3	2	3	1
3	3	3	2	2	1	3	2	3

Equivalently:

$\{1\}$	$\{2,3\}$	$\{4,5,6\}$
$\{2\}$	$\{3,4\}$	$\{1,5,6\}$
$\{3\}$	$\{4,5\}$	$\{1,2,6\}$
$\{4\}$	$\{5,6\}$	$\{1,2,3\}$
$\{5\}$	$\{1,6\}$	$\{2,3,4\}$
$\{6\}$	$\{1,2\}$	$\{3,4,5\}$
$\{1,3\}$	$\{2,5\}$	$\{4,6\}$
$\{2,4\}$	$\{3,6\}$	$\{1,5\}$
$\{3,5\}$	$\{1,4\}$	$\{2,6\}$

Disjoint set partitions

Disjoint set partitions

Fact

A $(\overline{1}, 1)$-locating array with N rows, k columns and v symbols is equivalent to k partitions of $\{1, \ldots, N\}$, each with v nonempty classes, such that no two of the kv classes are equal.

Disjoint set partitions

Fact
A $(\overline{1}, 1)$-locating array with N rows, k columns and v symbols is equivalent to k partitions of $\{1, \ldots, N\}$, each with v nonempty classes, such that no two of the kv classes are equal.

I'll say *disjoint v-partitions of $\{1, \ldots, N\}$* from now on.

Disjoint set partitions

Fact
A $(\overline{1}, 1)$-locating array with N rows, k columns and v symbols is equivalent to k partitions of $\{1, \ldots, N\}$, each with v nonempty classes, such that no two of the kv classes are equal.

I'll say *disjoint v-partitions of $\{1, \ldots, N\}$* from now on.

Our problem (rephrased)
Given N and v, find the maximum number of disjoint v-partitions of $\{1, \ldots, N\}$.

Disjoint set partitions

Fact
A $(\overline{1}, 1)$-locating array with N rows, k columns and v symbols is equivalent to k partitions of $\{1, \ldots, N\}$, each with v nonempty classes, such that no two of the kv classes are equal.

I'll say *disjoint v-partitions of $\{1, \ldots, N\}$* from now on.

Our problem (rephrased)
Given N and v, find the maximum number of disjoint v-partitions of $\{1, \ldots, N\}$.

Disjoint set partitions

Fact
A $(\overline{1}, 1)$-locating array with N rows, k columns and v symbols is equivalent to k partitions of $\{1, \ldots, N\}$, each with v nonempty classes, such that no two of the kv classes are equal.

I'll say *disjoint v-partitions of $\{1, \ldots, N\}$* from now on.

Our problem (rephrased)
Given N and v, find the maximum number of disjoint v-partitions of $\{1, \ldots, N\}$.

Disjoint set partitions

Fact
A $(\overline{1}, 1)$-locating array with N rows, k columns and v symbols is equivalent to k partitions of $\{1, \ldots, N\}$, each with v nonempty classes, such that no two of the kv classes are equal.

I'll say *disjoint v-partitions of $\{1, \ldots, N\}$* from now on.

Our problem (rephrased)
Given N and v, find the maximum number of disjoint v-partitions of $\{1, \ldots, N\}$.

Similar problems have been considered in the set systems literature, but not this one exactly.

Shapes

Shapes

$\{1\}$	$\{2, 3\}$	$\{4, 5, 6\}$
$\{2\}$	$\{3, 4\}$	$\{1, 5, 6\}$
$\{3\}$	$\{4, 5\}$	$\{1, 2, 6\}$
$\{4\}$	$\{5, 6\}$	$\{1, 2, 3\}$
$\{5\}$	$\{1, 6\}$	$\{2, 3, 4\}$
$\{6\}$	$\{1, 2\}$	$\{3, 4, 5\}$
$\{1, 3\}$	$\{2, 5\}$	$\{4, 6\}$
$\{2, 4\}$	$\{3, 6\}$	$\{1, 5\}$
$\{3, 5\}$	$\{1, 4\}$	$\{2, 6\}$

Shapes

partition			shape
$\{1\}$	$\{2,3\}$	$\{4,5,6\}$	[1,2,3]
$\{2\}$	$\{3,4\}$	$\{1,5,6\}$	[1,2,3]
$\{3\}$	$\{4,5\}$	$\{1,2,6\}$	[1,2,3]
$\{4\}$	$\{5,6\}$	$\{1,2,3\}$	[1,2,3]
$\{5\}$	$\{1,6\}$	$\{2,3,4\}$	[1,2,3]
$\{6\}$	$\{1,2\}$	$\{3,4,5\}$	[1,2,3]
$\{1,3\}$	$\{2,5\}$	$\{4,6\}$	[2,2,2]
$\{2,4\}$	$\{3,6\}$	$\{1,5\}$	[2,2,2]
$\{3,5\}$	$\{1,4\}$	$\{2,6\}$	[2,2,2]

Shapes

partition			shape
$\{1\}$	$\{2,3\}$	$\{4,5,6\}$	[1,2,3]
$\{2\}$	$\{3,4\}$	$\{1,5,6\}$	[1,2,3]
$\{3\}$	$\{4,5\}$	$\{1,2,6\}$	[1,2,3]
$\{4\}$	$\{5,6\}$	$\{1,2,3\}$	[1,2,3]
$\{5\}$	$\{1,6\}$	$\{2,3,4\}$	[1,2,3]
$\{6\}$	$\{1,2\}$	$\{3,4,5\}$	[1,2,3]
$\{1,3\}$	$\{2,5\}$	$\{4,6\}$	[2,2,2]
$\{2,4\}$	$\{3,6\}$	$\{1,5\}$	[2,2,2]
$\{3,5\}$	$\{1,4\}$	$\{2,6\}$	[2,2,2]

k disjoint v-partitions of $\{1,\ldots,N\}$ give rise to k *shapes*, each with v parts, such at most $\binom{N}{i}$ of the kv parts are equal to i for $i \in \{1,\ldots,N\}$.

Shapes

partition			shape
$\{1\}$	$\{2,3\}$	$\{4,5,6\}$	[1,2,3]
$\{2\}$	$\{3,4\}$	$\{1,5,6\}$	[1,2,3]
$\{3\}$	$\{4,5\}$	$\{1,2,6\}$	[1,2,3]
$\{4\}$	$\{5,6\}$	$\{1,2,3\}$	[1,2,3]
$\{5\}$	$\{1,6\}$	$\{2,3,4\}$	[1,2,3]
$\{6\}$	$\{1,2\}$	$\{3,4,5\}$	[1,2,3]
$\{1,3\}$	$\{2,5\}$	$\{4,6\}$	[2,2,2]
$\{2,4\}$	$\{3,6\}$	$\{1,5\}$	[2,2,2]
$\{3,5\}$	$\{1,4\}$	$\{2,6\}$	[2,2,2]

k disjoint v-partitions of $\{1,\ldots,N\}$ give rise to k *shapes*, each with v parts, such at most $\binom{N}{i}$ of the kv parts are equal to i for $i \in \{1,\ldots,N\}$.

I'll say *admissible family of v-shapes* from now on.

Shapes

partition			shape
$\{1\}$	$\{2,3\}$	$\{4,5,6\}$	[1,2,3]
$\{2\}$	$\{3,4\}$	$\{1,5,6\}$	[1,2,3]
$\{3\}$	$\{4,5\}$	$\{1,2,6\}$	[1,2,3]
$\{4\}$	$\{5,6\}$	$\{1,2,3\}$	[1,2,3]
$\{5\}$	$\{1,6\}$	$\{2,3,4\}$	[1,2,3]
$\{6\}$	$\{1,2\}$	$\{3,4,5\}$	[1,2,3]
$\{1,3\}$	$\{2,5\}$	$\{4,6\}$	[2,2,2]
$\{2,4\}$	$\{3,6\}$	$\{1,5\}$	[2,2,2]
$\{3,5\}$	$\{1,4\}$	$\{2,6\}$	[2,2,2]

k disjoint v-partitions of $\{1,\ldots,N\}$ give rise to k *shapes*, each with v parts, such at most $\binom{N}{i}$ of the kv parts are equal to i for $i \in \{1,\ldots,N\}$.

I'll say *admissible family of v-shapes* from now on.

Shapes

Shapes

Theorem
A family of disjoint partitions of $\{1, \ldots, N\}$ with specified shapes exists if and only if the family of specified shapes is admissible.

Shapes

Theorem
A family of disjoint partitions of $\{1, \ldots, N\}$ with specified shapes exists if and only if the family of specified shapes is admissible.

This is a generalisation of:

Baranyai's theorem
There are $\frac{1}{v}\binom{uv}{u}$ disjoint v-partitions of $\{1, \ldots, uv\}$ such that each class of each partition has size u.

Shapes

Theorem
A family of disjoint partitions of $\{1, \ldots, N\}$ with specified shapes exists if and only if the family of specified shapes is admissible.

This is a generalisation of:

Baranyai's theorem
There are $\frac{1}{v}\binom{uv}{u}$ disjoint v-partitions of $\{1, \ldots, uv\}$ such that each class of each partition has size u.

Our proof is adapted from an inductive proof Baranyai's theorem, due to Brouwer and Schrijver, that uses the integer flow theorem.

Shapes

Theorem
A family of disjoint partitions of $\{1, \ldots, N\}$ with specified shapes exists if and only if the family of specified shapes is admissible.

This is a generalisation of:

Baranyai's theorem
There are $\frac{1}{v}\binom{uv}{u}$ disjoint v-partitions of $\{1, \ldots, uv\}$ such that each class of each partition has size u.

Our proof is adapted from an inductive proof Baranyai's theorem, due to Brouwer and Schrijver, that uses the integer flow theorem.

See also Bahmanian and Bryant.

Shapes

Theorem
A family of disjoint partitions of $\{1, \ldots, N\}$ with specified shapes exists if and only if the family of specified shapes is admissible.

This is a generalisation of:

Baranyai's theorem
There are $\frac{1}{v}\binom{uv}{u}$ disjoint v-partitions of $\{1, \ldots, uv\}$ such that each class of each partition has size u.

Our proof is adapted from an inductive proof Baranyai's theorem, due to Brouwer and Schrijver, that uses the integer flow theorem.

See also Bahmanian and Bryant.

Our problem (re-rephrased)
Given N and v, find the maximum size of an admissible family of v-shapes.

Maximal families

Maximal families

Example: $\boldsymbol{N = 38, v = 7}$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.

Maximal families

Example: $\boldsymbol{N = 38, v = 7}$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

$$[1, 6, 6, 6, 6, 6, 7] \times \binom{38}{1}$$

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

$$[1, 6, 6, 6, 6, 6, 7] \times \binom{38}{1}$$
$$[2, 6, 6, 6, 6, 6, 6] \times \binom{38}{2}$$

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

$$[1, 6, 6, 6, 6, 6, 7] \times \binom{38}{1}$$
$$[2, 6, 6, 6, 6, 6, 6] \times \binom{38}{2}$$
$$[3, 5, 6, 6, 6, 6, 6] \times \binom{38}{3}$$

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

$$[1, 6, 6, 6, 6, 6, 7] \times \binom{38}{1}$$
$$[2, 6, 6, 6, 6, 6, 6] \times \binom{38}{2}$$
$$[3, 5, 6, 6, 6, 6, 6] \times \binom{38}{3}$$
$$[4, 5, 5, 6, 6, 6, 6] \times \binom{38}{4}$$

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

$$
\begin{aligned}
&[1, 6, 6, 6, 6, 6, 7] \times \binom{38}{1} \\
&[2, 6, 6, 6, 6, 6, 6] \times \binom{38}{2} \\
&[3, 5, 6, 6, 6, 6, 6] \times \binom{38}{3} \\
&[4, 5, 5, 6, 6, 6, 6] \times \binom{38}{4} \\
&[5, 5, 5, 5, 6, 6, 6] \times \lfloor (\binom{38}{5} - 2\binom{38}{4} - \binom{38}{3})/4 \rfloor
\end{aligned}
$$

Maximal families

Example: $N = 38$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 5,\ d = v(f+1) - N = 4)$

- There are at most $\binom{38}{1} + \binom{38}{2}$ *special* shapes containing a 1 or 2.
- Let the *defect* of a part $x \leqslant 5$ be $f + 1 - x = 6 - x$.
- The total defect in the other shapes is at most $3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5}$.
- Any shape has total defect at least $d = 4$.
- So there are at most $\lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ other shapes.
- So at most $\binom{38}{1} + \binom{38}{2} + \lfloor (3\binom{38}{3} + 2\binom{38}{4} + \binom{38}{5})/4 \rfloor$ shapes in total.

$$[1, 6, 6, 6, 6, 6, 7] \times \binom{38}{1}$$
$$[2, 6, 6, 6, 6, 6, 6] \times \binom{38}{2}$$
$$[3, 5, 6, 6, 6, 6, 6] \times \binom{38}{3}$$
$$[4, 5, 5, 6, 6, 6, 6] \times \binom{38}{4}$$
$$[5, 5, 5, 5, 6, 6, 6] \times \lfloor (\binom{38}{5} - 2\binom{38}{4} - \binom{38}{3})/4 \rfloor$$

Close to $\binom{38}{5}$ 5s, and fewer than $\binom{38}{6}$ 6s are used.

Maximal families

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$[1, 6, 6, 7, 7, 7, 7] \times \binom{41}{1}$$

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$[1, 6, 6, 7, 7, 7, 7] \times \binom{41}{1}$$
$$[2, 6, 6, 6, 7, 7, 7] \times \binom{41}{2}$$

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$[1,6,6,7,7,7,7] \times \binom{41}{1}$$
$$[2,6,6,6,7,7,7] \times \binom{41}{2}$$
$$[3,6,6,6,6,7,7] \times \binom{41}{3}$$

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$[1,6,6,7,7,7,7] \times \binom{41}{1}$$
$$[2,6,6,6,7,7,7] \times \binom{41}{2}$$
$$[3,6,6,6,6,7,7] \times \binom{41}{3}$$
$$[4,6,6,6,6,6,7] \times \binom{41}{4}$$

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $\boldsymbol{N = 41,\ v = 7}$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$[1, 6, 6, 7, 7, 7, 7] \times \binom{41}{1}$$
$$[2, 6, 6, 6, 7, 7, 7] \times \binom{41}{2}$$
$$[3, 6, 6, 6, 6, 7, 7] \times \binom{41}{3}$$
$$[4, 6, 6, 6, 6, 6, 7] \times \binom{41}{4}$$
$$[5, 6, 6, 6, 6, 6, 6] \times \binom{41}{5}$$

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$[1,6,6,7,7,7,7] \times \binom{41}{1}$$
$$[2,6,6,6,7,7,7] \times \binom{41}{2}$$
$$[3,6,6,6,6,7,7] \times \binom{41}{3}$$
$$[4,6,6,6,6,6,7] \times \binom{41}{4}$$
$$[5,6,6,6,6,6,6] \times \binom{41}{5}$$

Too many 6s.

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41$, $v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$\begin{array}{l} [1,6,6,7,7,7,7] \times \binom{41}{1} \\ [2,6,6,6,7,7,7] \times \binom{41}{2} \\ [3,6,6,6,6,7,7] \times \binom{41}{3} \\ [4,6,6,6,6,6,7] \times \binom{41}{4} \\ [5,6,6,6,6,6,6] \times \binom{41}{5} - 2r \\ [5,5,6,6,6,6,7] \times\ r \end{array}$$

where $r = 2\binom{41}{1} + 3\binom{41}{2} + 4\binom{41}{3} + 5\binom{41}{4}$.

Maximal families

The idea on the previous slide works unless $N \equiv v - 1 \pmod{v}$.

Example: $N = 41, v = 7$ $\quad (f = \lfloor \frac{N+1}{v} \rfloor = 6,\ d = v(f+1) - N = 8)$

$$\begin{aligned}
&[1,6,6,7,7,7,7] \times \tbinom{41}{1} \\
&[2,6,6,6,7,7,7] \times \tbinom{41}{2} \\
&[3,6,6,6,6,7,7] \times \tbinom{41}{3} \\
&[4,6,6,6,6,6,7] \times \tbinom{41}{4} \\
&[5,6,6,6,6,6,6] \times \tbinom{41}{5} - 2r \\
&[5,5,6,6,6,6,7] \times\ r
\end{aligned}$$

where $r = 2\binom{41}{1} + 3\binom{41}{2} + 4\binom{41}{3} + 5\binom{41}{4}$.

Exactly $\binom{41}{5}$ 5s, close to $\binom{41}{6}$ 6s, and fewer than $\binom{41}{7}$ 7s are used.

The solution

The solution

Theorem

Let N and v be integers such that $2 \leqslant v \leqslant N$. The maximum number of disjoint v-partitions of $\{1, \ldots, N\}$ is

$$\left\lfloor \frac{1}{d} \sum_{\substack{i=f-d+2 \\ i \geqslant 1}}^{f} (f+1-i) \binom{N}{i} \right\rfloor + \sum_{i=1}^{f-d+1} \binom{N}{i},$$

where $f = \lfloor \frac{N+1}{v} \rfloor$ and $d = v(f+1) - N$.

The solution

Theorem

Let N and v be integers such that $2 \leqslant v \leqslant N$. The maximum number of disjoint v-partitions of $\{1, \ldots, N\}$ is

$$\left\lfloor \frac{1}{d} \sum_{\substack{i=f-d+2 \\ i \geqslant 1}}^{f} (f+1-i)\binom{N}{i} \right\rfloor + \sum_{i=1}^{f-d+1} \binom{N}{i},$$

where $f = \lfloor \frac{N+1}{v} \rfloor$ and $d = v(f+1) - N$.

Or, this is the maximum number of columns in a $(\overline{1}, 1)$-locating array with N rows and v symbols.

The solution

Theorem

Let N and v be integers such that $2 \leqslant v \leqslant N$. The maximum number of disjoint v-partitions of $\{1, \ldots, N\}$ is

$$\left\lfloor \frac{1}{d} \sum_{\substack{i=f-d+2 \\ i \geqslant 1}}^{f} (f+1-i)\binom{N}{i} \right\rfloor + \sum_{i=1}^{f-d+1} \binom{N}{i},$$

where $f = \lfloor \frac{N+1}{v} \rfloor$ and $d = v(f+1) - N$.

Or, this is the maximum number of columns in a $(\overline{1}, 1)$-locating array with N rows and v symbols.

For fixed v, we have $N = \frac{v}{v \log v - (v-1)\log(v-1)} \log k + O(\log \log k)$.

Future work

Future work

- Dealing with multiple faults.

Future work

- Dealing with multiple faults.
- Higher strength locating arrays (faults caused by t-way interactions).

Future work

- Dealing with multiple faults.
- Higher strength locating arrays (faults caused by t-way interactions).
- Sperner partition systems (Meagher, Moura, Stevens).

That’s all.