Covering Arrays with Row Limit: Bounds and Constructions

Nevena Francetić
Supervised by Prof. P. Danziger and Prof. E. Mendelsohn

Discrete Maths Research Group
Monash University

September 22, 2014.

Covering arrays

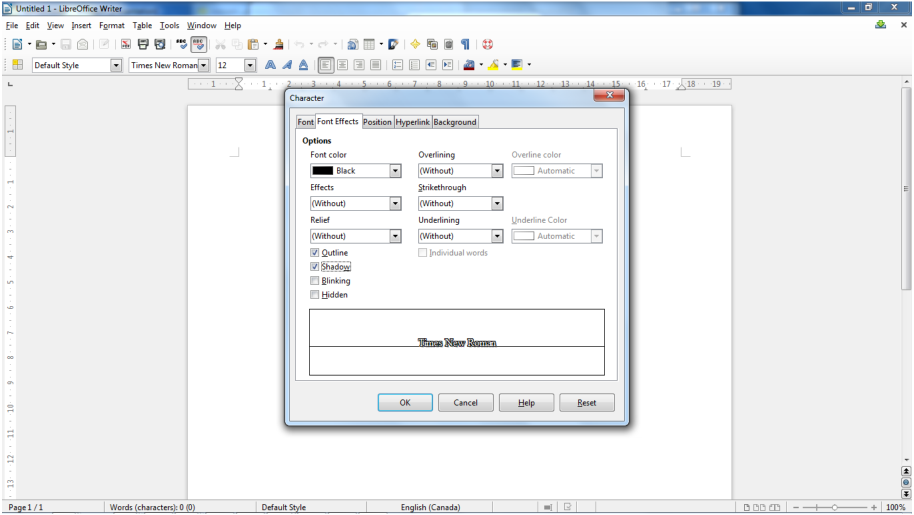
Covering arrays

- … is a test suite …

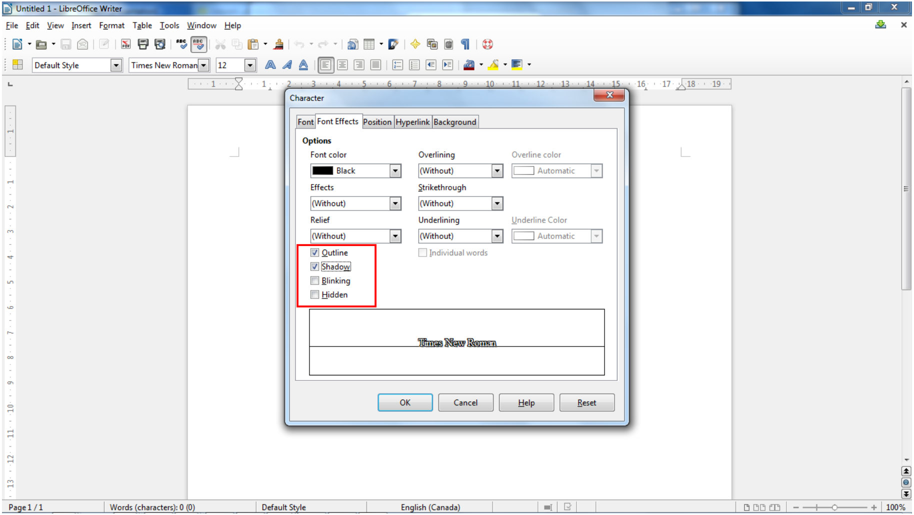
Covering arrays

- … is a test suite …
- … for verification of interactions between components.

Example

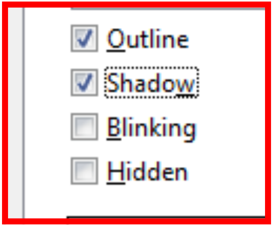

Untitled 1 - LibreOffice Writer
File Edit View Insert Format Table Tools Window Help
Default Style
Times New Roman
12
Character
Font
Font Effects
Position
Hyperlink
Background
Options
Font color
Black
Overlining
(Without)
Overline color
Automatic
Effects
(Without)
Strikethrough
(Without)
Relief
(Without)
Underlining
(Without)
Underline Color
Automatic
Outline
Shadow
Blinking
Hidden
Individual words
Times New Roman
OK
Cancel
Help
Reset
Page 1 / 1
Words (characters): 0 (0)
Default Style
English (Canada)
100%

Example

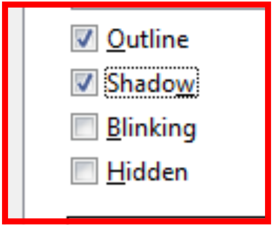


Untitled 1 - LibreOffice Writer
File Edit View Insert Format Table Tools Window Help
Default Style
Times New Roman
12
Character
Font
Font Effects
Position
Hyperlink
Background
Options
Font color
Black
Overlining
(Without)
Overline color
Automatic
Effects
(Without)
Strikethrough
(Without)
Relief
(Without)
Underlining
(Without)
Underline Color
Automatic
Outline
Shadow
Blinking
Hidden
Individual words
Times New Roman
OK
Cancel
Help
Reset
Page 1 / 1
Words (characters): 0 (0)
Default Style
English (Canada)
100%

Example

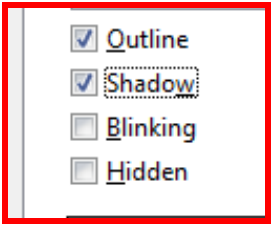


Example



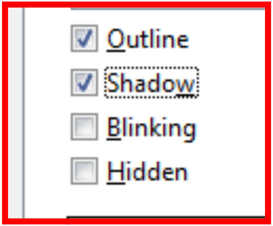
- Components: outline, shadow, blinking, hidden

Example



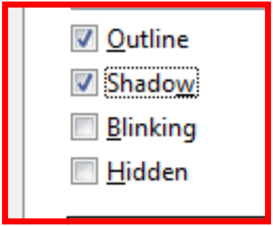
- Components: outline, shadow, blinking, hidden
- Levels for each component: present or absent (1 or 0)

Example



- Components: outline, shadow, blinking, hidden
- Levels for each component: present or absent (1 or 0)
- What interactions? Pairwise interactions.

Example



- Components: outline, shadow, blinking, hidden
- Levels for each component: present or absent (1 or 0)
- What interactions? Pairwise interactions.

Testing two at time: it would take $\binom{4}{2}2^2 = 24$ tests.

Example

Outline	Shadow	Blinking	Hidden
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

Example

Outline	Shadow	Blinking	Hidden
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

Example

Outline	Shadow	Blinking	Hidden
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

Testing can be done in only 5 iterations.

Example

Outline	Shadow	Blinking	Hidden
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

Parameters of a Covering Array

A covering array is characterized by:

- k: the number of components (columns)
- v: the number of levels for each component (alphabet size)
- t: strength $\Rightarrow$ testing interactions between t columns

Parameters of a Covering Array

A covering array is characterized by:

- k: the number of components (columns)
- v: the number of levels for each component (alphabet size)
- t: strength $\Rightarrow$ testing interactions between t columns

Goal:

- find the smallest number of rows, called size N of a covering array. Size of an optimal CA is usually denoted by $CAN(t, k, v)$.

Some facts about covering arrays

Some facts about covering arrays

- $N = \mathcal{O}(\log k))$ (Gargano et al., 1993)

Some facts about covering arrays

- $N = \mathcal{O}(\log k))$ (Gargano et al., 1993)
- The exact size of a covering array is ONLY known for one family of arrays (Kleitman and Spencer, 1973; Katona, 1973):

$$CA(N;2,k,2) \text{ exists for all } k \leq \binom{N-1}{\lfloor \frac{N}{2} - 1 \rfloor}$$

Some facts about covering arrays

- $N = \mathcal{O}(\log k))$ (Gargano et al., 1993)
- The exact size of a covering array is ONLY known for one family of arrays (Kleitman and Spencer, 1973; Katona, 1973):

$$CA(N; 2, k, 2) \text{ exists for all } k \leq \binom{N-1}{\lfloor \frac{N}{2} - 1 \rfloor}$$

- Finding an optimal covering array is NP-complete when extra constraints are imposed even when $v = 2$ (Maltais and Moura, 2011).

More on covering arrays

`http://www.pairwise.org/tools.asp` contains:

- 39 software tools for constructing *CA*s
- both commercial and open source

Covering array with row limit (*CARL*)

Covering array with row limit (*CARL*)

$CARL(N = 12; t = 2, k = 6, v = 2 : w = 4)$

0	0	–	1	–	0
0	–	1	1	1	–
0	1	0	–	–	1
0	–	0	0	0	–
1	0	1	0	–	–
1	–	0	–	1	0
1	–	1	–	0	1
1	1	–	1	–	1
–	0	0	1	0	–
–	1	1	–	1	0
–	0	–	0	1	1
–	1	–	0	0	0

Covering array with row limit (*CARL*)

$CARL(N = 12; t = 2, k = 6, v = 2 : w = 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
3	0	1	0	–	–	1
4	0	–	0	0	0	–
5	1	0	1	0	–	–
6	1	–	0	–	1	0
7	1	–	1	–	0	1
8	1	1	–	1	–	1
9	–	0	0	1	0	–
10	–	1	1	–	1	0
11	–	0	–	0	1	1
12	–	1	–	0	0	0

- k = the number of columns (components)

Covering array with row limit (*CARL*)

$CARL(N = 12; t = 2, k = 6, v = 2 \colon w = 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
3	0	1	0	–	–	1
4	0	–	0	0	0	–
5	1	0	1	0	–	–
6	1	–	0	–	1	0
7	1	–	1	–	0	1
8	1	1	–	1	–	1
9	–	0	0	1	0	–
10	–	1	1	–	1	0
11	–	0	–	0	1	1
12	–	1	–	0	0	0

- k = the number of columns (components)
- v = the alphabet size; the number of different values assigned to a column

Covering array with row limit (*CARL*)

$CARL(N = 12; t = 2, k = 6, v = 2 : w = 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
3	0	1	0	–	–	1
4	0	–	0	0	0	–
5	1	0	1	0	–	–
6	1	–	0	–	1	0
7	1	–	1	–	0	1
8	1	1	–	1	–	1
9	–	0	0	1	0	–
10	–	1	1	–	1	0
11	–	0	–	0	1	1
12	–	1	–	0	0	0

- k = the number of columns (components)
- v = the alphabet size; the number of different values assigned to a column
- w = row limit; the number of non-empty cells in a row

Covering array with row limit (*CARL*)

$CARL(N = 12; t = 2, k = 6, v = 2 : w = 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
3	0	1	0	–	–	1
4	0	–	0	0	0	–
5	1	0	1	0	–	–
6	1	–	0	–	1	0
7	1	–	1	–	0	1
8	1	1	–	1	–	1
9	–	0	0	1	0	–
10	–	1	1	–	1	0
11	–	0	–	0	1	1
12	–	1	–	0	0	0

- k = the number of columns (components)
- v = the alphabet size; the number of different values assigned to a column
- w = row limit; the number of non-empty cells in a row
- t = strength

Covering array with row limit (*CARL*)

$CARL(N = 12; t = 2, k = 6, v = 2 : w = 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
3	0	1	0	–	–	1
4	0	–	0	0	0	–
5	1	0	1	0	–	–
6	1	–	0	–	1	0
7	1	–	1	–	0	1
8	1	1	–	1	–	1
9	–	0	0	1	0	–
10	–	1	1	–	1	0
11	–	0	–	0	1	1
12	–	1	–	0	0	0

- k = the number of columns (components)
- v = the alphabet size; the number of different values assigned to a column
- w = row limit; the number of non-empty cells in a row
- t = strength
- N = size; goal find minimum N

Covering arrays vs *CARL*s

$$CAN(t, k, v) \leq CARLN(t, k, v\colon w(k))$$

Covering arrays vs *CARL*s

$$CAN(t, k, v) \leq CARLN(t, k, v : w(k))$$

Theorem (Gargano et al.(1993))

$$\limsup_{k \to \infty} CAN(2, k, v) = \Theta(\log k).$$

Covering arrays vs *CARL*s

$$CAN(t,k,v) \leq CARLN(t,k,v: w(k))$$

Theorem (Gargano et al.(1993))

$$\limsup_{k\to\infty} CAN(2,k,v) = \Theta(\log k).$$

$$\Omega(\log k) = CAN(2,k,v) \leq CAN(t,k,v) \leq CARLN(t,k,v: w(k))$$

Schönheim lower bound

Theorem

$$CARLN_{\lambda}(t, k, v : w) \geq SB(t, k, v : w)$$

$$SB(t, k, v : w) = \left\lceil \frac{vk}{w} \left\lceil \frac{v(k-1)}{w-1} \cdots \left\lceil \frac{v(k-t+1)}{w-t+1} \right\rceil \cdots \right\rceil \right\rceil.$$

$CARL(12; 2, 6, 2 : 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
3	0	1	0	–	–	1
4	0	–	0	0	0	–
5	1	0	1	0	–	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Schönheim lower bound

- $w(k) = c$, $c \in \mathbb{N}$:

$$CARLN(t, k, v\colon w) = \frac{\binom{k}{t}}{\binom{w}{t}} v^t (1 + \mathrm{o}(1))$$

Schönheim lower bound

- $w(k) = c$, $c \in \mathbb{N}$:

$$CARLN(t, k, v\colon w) = \frac{\binom{k}{t}}{\binom{w}{t}} v^t (1 + \mathrm{o}(1))$$

$$CARLN(t, k, v\colon w) = \Theta(k^t)$$

Schönheim lower bound

- $w(k) = c$, $c \in \mathbb{N}$:

$$CARLN(t, k, v\colon w) = \frac{\binom{k}{t}}{\binom{w}{t}} v^t (1 + \mathrm{o}(1))$$

$$CARLN(t, k, v\colon w) = \Theta(k^t)$$

- $\lim_{k \to \infty} \frac{k^t}{w(k)^t \log k} = 0$

$$\lim_{k \to \infty} \frac{SB(t, k, v\colon w(k))}{\log k} = 0$$

A probabilistic upper bound

$$CARL(t = 2, k = 6, v = 2 \colon w = 4)$$

1	2	3	4	5	6
0	0	0	0	—	—
0	0	0	1	—	—
0	0	1	0	—	—
0	0	1	1	—	—
⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	—	0	—
0	0	0	—	1	—
0	0	1	—	0	—
0	0	1	—	1	—
⋮	⋮	⋮	⋮	⋮	⋮
—	0	1	—	0	0
—	0	1	—	0	1
—	0	1	—	1	0
—	0	1	—	1	1
⋮	⋮	⋮	⋮	⋮	⋮
—	—	1	1	0	0
—	—	1	1	0	1
—	—	1	1	1	0
—	—	1	1	1	1

1	2	3	4	5	6

A probabilistic upper bound

$$CARL(t=2, k=6, v=2: w=4)$$

1	2	3	4	5	6	
0	0	0	0	–	–	IN
0	0	0	1	–	–	
0	0	1	0	–	–	
0	0	1	1	–	–	
⋮	⋮	⋮	⋮	⋮	⋮	
0	0	0	–	0	–	
0	0	0	–	1	–	
0	0	1	–	0	–	
0	0	1	–	1	–	
⋮	⋮	⋮	⋮	⋮	⋮	
–	0	1	–	0	0	
–	0	1	–	0	1	
–	0	1	–	1	0	
–	0	1	–	1	1	
⋮	⋮	⋮	⋮	⋮	⋮	
–	–	1	1	0	0	
–	–	1	1	0	1	
–	–	1	1	1	0	
–	–	1	1	1	1	

1	2	3	4	5	6
0	0	0	0	–	–

A probabilistic upper bound

$$CARL(t = 2, k = 6, v = 2 \colon w = 4)$$

1	2	3	4	5	6	
0	0	0	0	–	–	
0	0	0	1	–	–	OUT
0	0	1	0	–	–	
0	0	1	1	–	–	
⋮	⋮	⋮	⋮	⋮	⋮	
0	0	0	–	0	–	
0	0	0	–	1	–	
0	0	1	–	0	–	
0	0	1	–	1	–	
⋮	⋮	⋮	⋮	⋮	⋮	
–	0	1	–	0	0	
–	0	1	–	0	1	
–	0	1	–	1	0	
–	0	1	–	1	1	
⋮	⋮	⋮	⋮	⋮	⋮	
–	–	1	1	0	0	
–	–	1	1	0	1	
–	–	1	1	1	0	
–	–	1	1	1	1	

1	2	3	4	5	6
0	0	0	0	–	–

A probabilistic upper bound

$$CARL(t = 2, k = 6, v = 2 \colon w = 4)$$

1	2	3	4	5	6	
0	0	0	0	–	–	
0	0	0	1	–	–	
0	0	1	0	–	–	OUT
0	0	1	1	–	–	
⋮	⋮	⋮	⋮	⋮	⋮	
0	0	0	–	0	–	
0	0	0	–	1	–	
0	0	1	–	0	–	
0	0	1	–	1	–	
⋮	⋮	⋮	⋮	⋮	⋮	
–	0	1	–	0	0	
–	0	1	–	0	1	
–	0	1	–	1	0	
–	0	1	–	1	1	
⋮	⋮	⋮	⋮	⋮	⋮	
–	–	1	1	0	0	
–	–	1	1	0	1	
–	–	1	1	1	0	
–	–	1	1	1	1	

1	2	3	4	5	6
0	0	0	0	–	–

A probabilistic upper bound

$$CARL(t = 2, k = 6, v = 2 \colon w = 4)$$

1	2	3	4	5	6	
0	0	0	0	–	–	
0	0	0	1	–	–	
0	0	1	0	–	–	
0	0	1	1	–	–	IN
⋮	⋮	⋮	⋮	⋮	⋮	
0	0	0	–	0	–	
0	0	0	–	1	–	
0	0	1	–	0	–	
0	0	1	–	1	–	
⋮	⋮	⋮	⋮	⋮	⋮	
–	0	1	–	0	0	
–	0	1	–	0	1	
–	0	1	–	1	0	
–	0	1	–	1	1	
⋮	⋮	⋮	⋮	⋮	⋮	
–	–	1	1	0	0	
–	–	1	1	0	1	
–	–	1	1	1	0	
–	–	1	1	1	1	

1	2	3	4	5	6
0	0	0	0	–	–
0	0	1	1	–	–

A probabilistic upper bound

$$CARL(t = 2, k = 6, v = 2\colon\ w = 4)$$

1	2	3	4	5	6	
0	0	0	0	–	–	
0	0	0	1	–	–	
0	0	1	0	–	–	
0	0	1	1	–	–	
⋮	⋮	⋮	⋮	⋮	⋮	
0	0	0	–	0	–	
0	0	0	–	1	–	
0	0	1	–	0	–	
0	0	1	–	1	–	
⋮	⋮	⋮	⋮	⋮	⋮	
–	0	1	–	0	0	
–	0	1	–	0	1	
–	0	1	–	1	0	
–	0	1	–	1	1	
⋮	⋮	⋮	⋮	⋮	⋮	
–	–	1	1	0	0	
–	–	1	1	0	1	
–	–	1	1	1	0	
–	–	1	1	1	1	IN

1	2	3	4	5	6
0	0	0	0	–	–
0	0	1	1	–	–
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
–	–	1	1	1	1

A probabilistic upper bound

$$CARL(t = 2, k = 6, v = 2 \colon w = 4)$$

1	2	3	4	5	6
0	0	0	0	—	—
0	0	0	1	—	—
0	0	1	0	—	—
0	0	1	1	—	—
⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	—	0	—
0	0	0	—	1	—
0	0	1	—	0	—
0	0	1	—	1	—
⋮	⋮	⋮	⋮	⋮	⋮
—	0	1	—	0	0
—	0	1	—	0	1
—	0	1	—	1	0
—	0	1	—	1	1
⋮	⋮	⋮	⋮	⋮	⋮
—	—	1	1	0	0
—	—	1	1	0	1
—	—	1	1	1	0
—	—	1	1	1	1

1	2	3	4	5	6
0	0	0	0	—	—
0	0	1	1	—	—
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
—	—	1	1	1	1

For each uncovered pair, add a row to cover it.

A probabilistic upper bound (UB_1)

Theorem (N.F., Danziger, Mendelsohn)

Let $c_1, c_2 > 1$ such that $\frac{1}{c_1} + \frac{1}{c_2} < 1$. Then,

$$CARLN(t,k,v\colon w) \leq UB_1(t,k,v\colon w)$$

$$UB_1(t,k,v\colon w) = c_1 \frac{\binom{k}{t}}{\binom{w}{t}} v^t \left(1 + \ln \frac{c_2}{c_1}\binom{w}{t}\right)$$

Asymptotic size of UB_1

Theorem (N.F., Danziger, Mendelsohn)

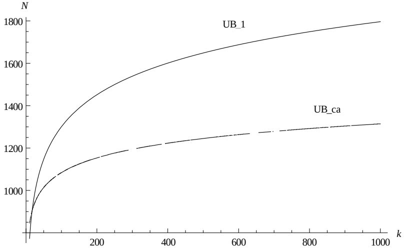
If $w(k) = \Theta(k)$, *then*

$$\limsup_{k \to \infty} CARLN(t, k, v \colon w(k)) = \Theta(\log k)$$

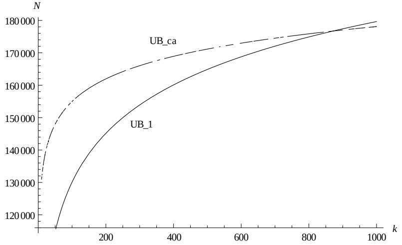
UB_1 for covering arrays

Theorem (Godbole et al., 1996)

$$CAN(t,k,v) \leq \frac{-\ln\left(ev^t t\binom{k-1}{t-1}\right)}{\ln\left(1-\frac{1}{v^t}\right)} = UB_{ca}(t,k,v).$$



$UB_1(2,k,10\colon k)$, $UB_{ca}(2,k,10)$



$UB_1(2,k,100\colon k)$, $UB_{ca}(2,k,100)$

Improvement to UB_1

Theorem (N.F., Danziger, Mendelsohn)

If $w(k) \ln w(k) = \mathrm{o}(k)$, *then*

$$CARLN(t,k,v\colon w(k)) \leq \frac{\binom{k}{t}}{\binom{w}{t}} v^t \left(1 + \ln \binom{w}{t}\right)(1+\mathrm{o}(1)).$$

Summary

$w = const$	Schönheim bound $CARLN = \Theta(k^t)$	constructive
$w(k) \ln w(k) = \mathrm{o}(k)$	improved UB_1	constructive
$\lim_{k\to\infty} \frac{k^t}{w(k)^t \log k} \neq 0$	Schöheim bound is $\Omega(\log k)$	
any $w(k)$	UB_1	not constructive
$w(k) = \Theta(k)$	$CARLN = \Theta(\log k)$	

Greedy algorithm

A generalization of *AETG* algorithm for covering arrays (Cohen et al., 1997).

Greedy algorithm

A generalization of *AETG* algorithm for covering arrays (Cohen et al., 1997).

Algorithm:

- Build an array one row at a time.

Greedy algorithm

A generalization of *AETG* algorithm for covering arrays (Cohen et al., 1997).

Algorithm:

- Build an array one row at a time.
- At each step compute the average number of uncovered t-tuples contained in any admissible row.

Greedy algorithm

A generalization of *AETG* algorithm for covering arrays (Cohen et al., 1997).

Algorithm:

- Build an array one row at a time.
- At each step compute the average number of uncovered t-tuples contained in any admissible row.
- Add a row which covers at least the average number of uncovered t-tuples.

Greedy algorithm

A generalization of *AETG* algorithm for covering arrays (Cohen et al., 1997).

Algorithm:

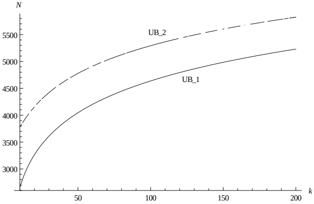
- Build an array one row at a time.
- At each step compute the average number of uncovered t-tuples contained in any admissible row.
- Add a row which covers at least the average number of uncovered t-tuples.

Theorem (N.F., Danziger, Mendelsohn)

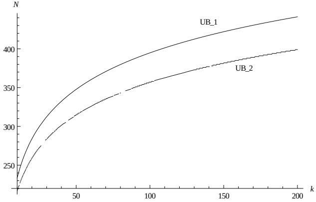
$$CARLN(t,k,v\colon w(k)) \leq UB_2(t,k,v\colon w)$$

$$UB_2(t,k,v\colon w) = \left\lceil 1 - \frac{\ln\left(\binom{k}{t}v^t - \binom{w}{t}\right)}{\ln\left(1 - \frac{\binom{w}{t}}{\binom{k}{t}v^t}\right)} \right\rceil .$$

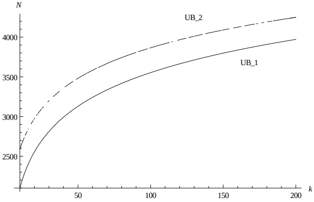
UB_1 vs UB_2



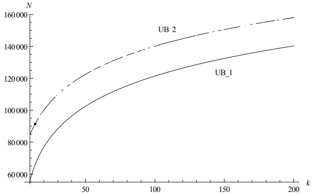
$UB_1(2, k, 10 \colon k/2)$, $UB_2(2, k, 10 \colon k/2)$



$UB_1(2, k, 5 \colon 9k/10)$, $UB_2(2, k, 5 \colon 9k/10)$



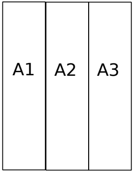
$UB_1(2, k, 15 \colon 9k/10)$, $UB_2(2, k, 15 \colon 9k/10)$



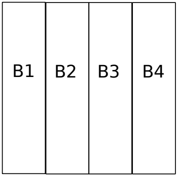
$UB_1(5, k, 5 \colon 9k/10)$, $UB_2(5, k, 5 \colon 9k/10)$

Product construction for strength $t = 2$

A:
$CARL(N_1; 2, k_1, v : w_1)$



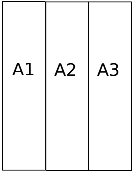
B:
$CARL(N_2; 2, k_2, v : w_2)$



Product construction for strength $t = 2$

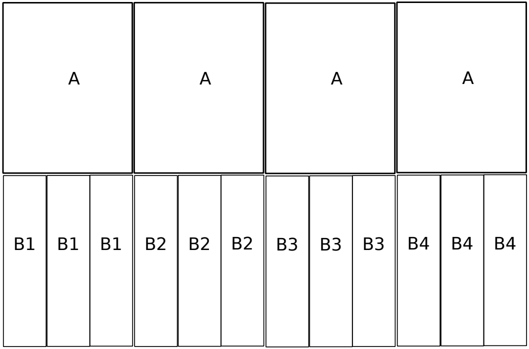
A:

$CARL(N_1; 2, k_1, v : w_1)$



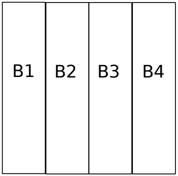
$A \times B$:

$CARL(N_1 + N_2; 2, k_1 k_2, v : \max\{k_1 w_2, k_2 w_1\})$



B:

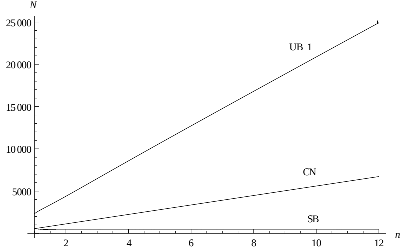
$CARL(N_2; 2, k_2, v : w_2)$



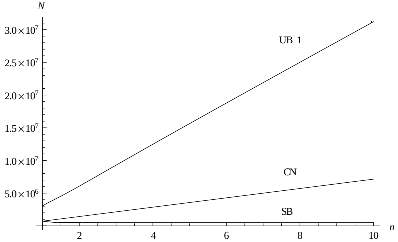
Product construction for strength $t = 2$

- Preserves the ration $\frac{w(k)}{k}$
- Retains logarithmic growth

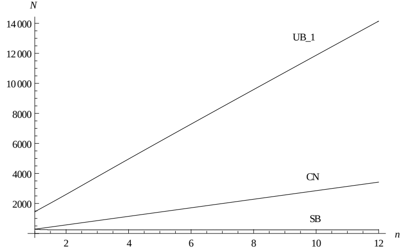
Constructed *CARLs*



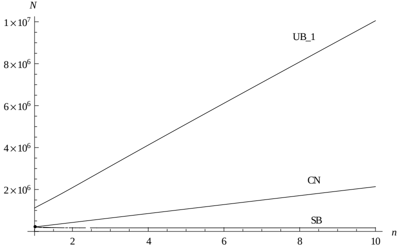
$UB_1(2, 12^n, 5: 3 \cdot 12^{n-1})$,
$CN(n) = nSB(2, 12, 5: 3)$,
$SB(2, 12^n, 5: 3 \cdot 12^{n-1})$



$UB_1(2, 21^n, 101: 3 \cdot 21^{n-1})$,
$CN(n) = nSB(2, 21, 101: 3)$,
$SB(2, 21^n, 101: 3 \cdot 21^{n-1})$

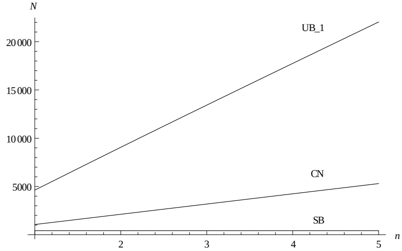


$UB_1(2, 12^n, 5: 4 \cdot 12^{n-1})$,
$CN(n) = nSB(2, 12, 5: 4)$,
$SB(2, 12^n, 5: 4 \cdot 12^{n-1})$

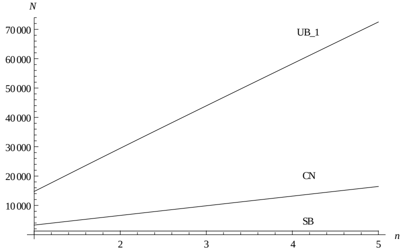


$UB_1(2, 17^n, 97: 4 \cdot 17^{n-1})$,
$CN(n) = nSB(2, 17, 97: 4)$,
$SB(2, 17^n, 97: 4 \cdot 17^{n-1})$

Constructed *CARLs*



$UB_1(2, 12^n16^n, 5\colon 3 \cdot 12^{n-1}16^n)$,
$CN(n) = nSB(2, 12, 5\colon 3) + nSB(2, 16, 6\colon 4)$,
$SB(2, 12^n16^n, 5\colon 3 \cdot 12^{n-1}16^n)$



$UB_1(2, 15^n20^n, 7\colon 3 \cdot 15^{n-1}20^n)$,
$CN(n) = nSB(2, 15, 7\colon 3) + nSB(2, 20, 7\colon 4)$
$SB(2, 15^n20^n, 7\colon 3 \cdot 15^{n-1}20^n)$

Constructions

- Wilson's Construction

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

$$\begin{aligned}\mathcal{B} \;=\; & \{\{1,2,3\},\ \{1,4,5\},\\ & \{1,3,6\},\ \{2,4,6\},\\ & \{2,5,6\},\ \{3,4,5\}\}\end{aligned}$$

Constructions

- Wilson's Construction

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

$$\begin{aligned}\mathcal{B} \;\; = \;\; \{&\{1,2,3\},\ \{1,4,5\},\\ &\{1,3,6\},\ \{2,4,6\},\\ &\{2,5,6\},\ \{3,4,5\}\}\end{aligned}$$

CARL(2,3,v: w)

C1	C2	C3

Constructions

- Wilson's Construction

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

$$\mathcal{B} = \{\{1, 2, 3\},\ \{1, 4, 5\},\ \{1, 3, 6\},\ \{2, 4, 6\},\ \{2, 5, 6\},\ \{3, 4, 5\}\}$$

CARL(2,3,v: w)

C1	C2	C3

1	2	3	4	5	6
C1	C2	C3			
C1			C2	C3	
C1		C2			C3
	C1		C2		C3
	C1			C2	C3
		C1	C2	C3	

Wilson's Construction

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

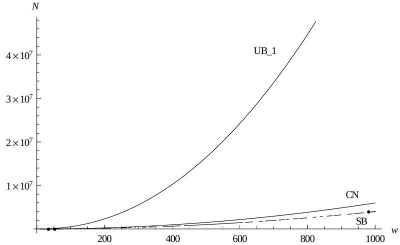
$$\mathcal{B} = \{\{1, 2, 3\},\ \{1, {\color{red}4, 5}\},\ \{1, 3, 6\},\ \{2, 4, 6\},\ \{2, 5, 6\},\ \{3, {\color{red}4, 5}\}\}$$

CARL(2,3,v: w)

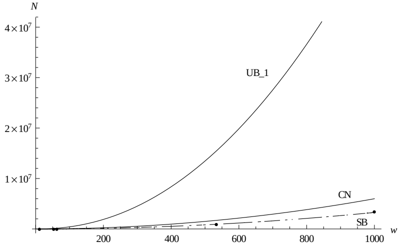
C1	C2	C3

1	2	3	4	5	6
C1	C2	C3			
C1			C2	C3	
C1		C2			C3
	C1		C2		C3
	C1			C2	C3
		C1	C2	C3	

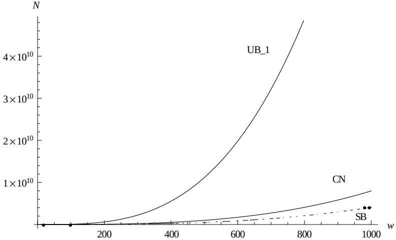
Wilson's construction using orthogonal arrays



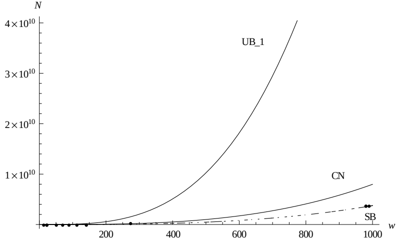
$UB_1(2, 2w, w-1\colon w)$,
$CN(w) = 6(w-1)^2$,
$SB(2, 2w, w-1\colon w)$



$UB_1(2, 1.81\,w, w-1\colon w)$
$CN(w) = 6(w-1)^2$,
$SB(2, 1.81\,w, w-1\colon w)$

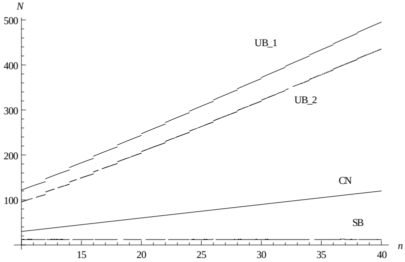


$UB_1(3, 8w/5, w-1\colon w)$,
$CN(w) = 8(w-1)^3$,
$SB(3, 8w/5, w-1\colon w)$

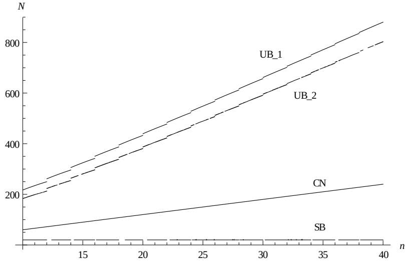


$UB_1(3, 17.1w/11, w-1\colon w)$
$CN(w) = 8(w-1)^3$,
$SB(3, 17.1w/11, w-1\colon w)$

Wilson's construction using covering arrays with $t = 2$ and $v = 2$



$UB_1(2, 3w(n)/2, 2\colon w(n))$, $UB_2(2, 3w(n)/2, 2\colon w(n))$, $CN(n) = 3n$, $SB(2, 3w(n)/2, 2\colon w(n))$



$UB_1(2, 2w(n), 2\colon w(n))$, $UB_2(2, 2w(n), 2\colon w(n))$, $CN(n) = 6n$, $SB(2, 2w(n), 2\colon w(n))$

$$w(n) = \binom{n-1}{\lfloor n/2 \rfloor - 1}$$

Summary

$w = const$	Schönheim bound $CARLN = \Theta(k^t)$	constructive
$w(k) \ln w(k) = o(k)$	improved UB_1	constructive
$\lim_{k\to\infty} \frac{k^t}{w(k)^t \log k} \neq 0$	Schöheim bound is $\Omega(\log k)$	
any $w(k)$	UB_1 UB_2	not constructive constructive
$w(k) = \Theta(k)$	$CARLN = \Theta(\log k)$	

*CARL*s with smaller size are more likely to be obtained through direct constructions.

▸ The End?

*CARL*s with constant row limit and $t = 2$

*CARL*s with constant row limit and $t = 2$

- Schönheim lower bound:

$$CARLN_\lambda(t, k, v : w) \geq \left\lceil \frac{vk}{w} \left\lceil \frac{v(k-1)}{w-1} \right\rceil \right\rceil.$$

- $w(k) = c$, $c \in \mathbb{N}$:

$$CARLN(t, k, v : w) = \frac{\binom{k}{t}}{\binom{w}{t}} v^t (1 + o(1))$$

- $$CARLN(t, k, v : w) = \Theta(k^t)$$

Back to definition

$CARL(N = 12; t = 2, k = 6, v = 2 : w = 4)$

0	0	–	1	–	0
0	–	1	1	1	–
0	1	0	–	–	1
0	–	0	0	0	–
1	0	1	0	–	–
1	–	0	–	1	0
1	–	1	–	0	1
1	1	–	1	–	1
–	0	0	1	0	–
–	1	1	–	1	0
–	0	–	0	1	1
–	1	–	0	0	0

Group divisible covering designs...

- ... are *CARL*s with $t = 2$ and constant w

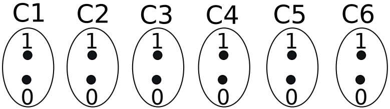
Group divisible covering designs...

- ... are *CARL*s with $t = 2$ and constant w
- $k - GDCD$ of type g^u, $(V, \mathcal{G}, \mathcal{B})$:

Group divisible covering designs...

- ... are *CARL*s with $t = 2$ and constant w
- $k - GDCD$ of type g^u, $(V, \mathcal{G}, \mathcal{B})$:

$|V| = 12$, $\mathcal{G} = \{C1, C2, \cdots, C6\}$



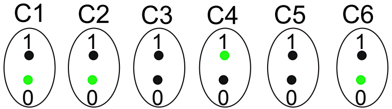
$CARL(12; 2, 6, 2 : 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Group divisible covering designs...

- ... are *CARL*s with $t = 2$ and constant w
- $k - GDCD$ of type g^u, $(V, \mathcal{G}, \mathcal{B})$:

$|V| = 12$, $\mathcal{G} = \{C1, C2, \cdots, C6\}$



$\{(1,0), (2,0), (4,1), (6,0)\}$

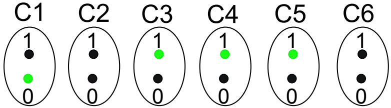
$CARL(12; 2, 6, 2: 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Group divisible covering designs...

- ... are *CARL*s with $t = 2$ and constant w
- $k - GDCD$ of type g^u, $(V, \mathcal{G}, \mathcal{B})$:

$|V| = 12$, $\mathcal{G} = \{C1, C2, \cdots, C6\}$



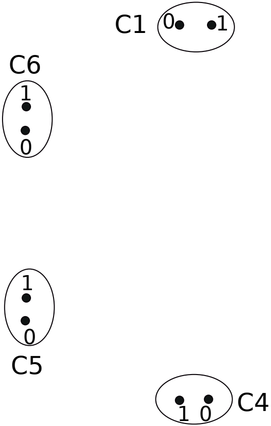
$\{(1,0),(2,0),(4,1),(6,0)\}$
$\{(1,0),(3,1),(4,1),(5,1)\}$

$CARL(12; 2, 6, 2 : 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Edge covering

- graph covering of a $K_{\underbrace{g,g,\ldots,g}_{u}}$ by K_k:

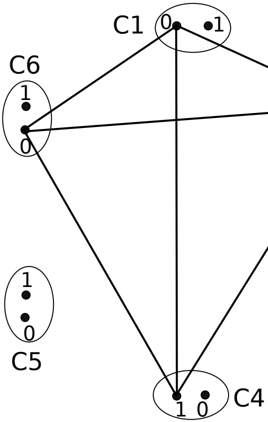


$CARL(12; 2, 6, 2: 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Edge covering

- graph covering of a $K_{\underbrace{g,g,\ldots,g}_{u}}$ by K_k:

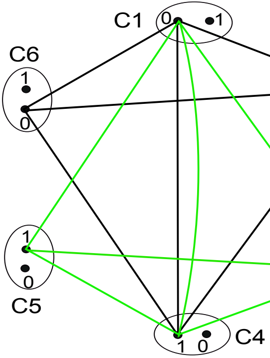


$CARL(12; 2, 6, 2 : 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Edge covering

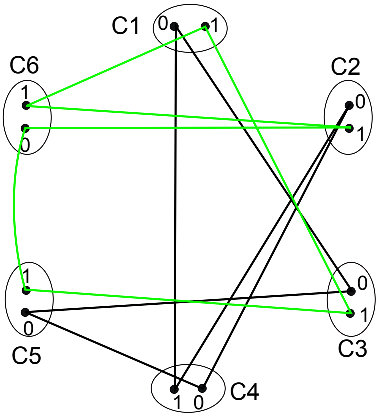
- graph covering of a $K_{\underbrace{g,g,\ldots,g}_{u}}$ by K_k:



$CARL(12; 2, 6, 2 : 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Excess graph when $t = 2$



$2C_6$

$CARL(12; 2, 6, 2 : 4)$

	1	2	3	4	5	6
1	0	0	–	1	–	0
2	0	–	1	1	1	–
3	0	1	0	–	–	1
4	0	–	0	0	0	–
5	1	0	1	0	–	–
6	1	–	0	–	1	0
7	1	–	1	–	0	1
8	1	1	–	1	–	1
9	–	0	0	1	0	–
10	–	1	1	–	1	0
11	–	0	–	0	1	1
12	–	1	–	0	0	0

Group divisible covering designs

- $C(k, g^u) = CARLN(2, u, g : k)$

Group divisible covering designs

- $C(k, g^u) = CARLN(2, u, g : k)$

- $k = 3$ done (Heinrich and Yin (1999))
- $k = 4$:

Group divisible covering designs

- $C(k, g^u) = CARLN(2, u, g\colon k)$
- $k = 3$ done (Heinrich and Yin (1999))
- $k = 4$:
  - $C(4, g^u) \geq \left\lceil \frac{gu}{4} \left\lceil \frac{g(u-1)}{3} \right\rceil \right\rceil$

Group divisible covering designs

- $C(k, g^u) = CARLN(2, u, g\colon k)$
- $k = 3$ done (Heinrich and Yin (1999))
- $k = 4$:
  - $C(4, g^u) \geq \left\lceil \frac{gu}{4} \left\lceil \frac{g(u-1)}{3} \right\rceil \right\rceil$
  - Construction methods: Wilson's construction, double group divisible designs, and some others

Group divisible covering designs

- $C(k, g^u) = CARLN(2, u, g \colon k)$
- $k = 3$ done (Heinrich and Yin (1999))
- $k = 4$:
  - $C(4, g^u) \geq \left\lceil \frac{gu}{4} \left\lceil \frac{g(u-1)}{3} \right\rceil \right\rceil$
  - Construction methods: Wilson's construction, double group divisible designs, and some others
  - Two types of objects: essential and auxiliary

$4-GDCDs$

Theorem

There exists a small positive integer δ, such that for any positive integer g and $u \geq 4$,

$$C(4, g^u) \leq \left\lceil \frac{gu}{4} \left\lceil \frac{g(u-1)}{3} \right\rceil \right\rceil + \delta,$$

except possibly when (1) $g = 17$ and $u \equiv 0 \pmod 3$, or (2) $g \geq 8$, $g \equiv 2, 5 \pmod 6$, and $u \equiv 23 \pmod{24}$ or $u \in \{29, 35, 41\}$.

Optimal $4-GDCDs$

Theorem

$C(4, g^u) = \left\lceil \frac{gu}{4} \left\lceil \frac{g(u-1)}{3} \right\rceil \right\rceil$ *when* $u \geq 4$ *and one of the following holds:*

- $u \equiv 0 \pmod{12}$ *except possibly when* $g = 17$, *or when* $u = 36$ *and* $g \equiv 5 \pmod{6}$, *or when* $(g, u) = (11, 24)$;
- $u \equiv 1, 4 \pmod{12}$, *except when* $(g, u) \in \{(2, 4), (6, 4)\}$;
- $u \equiv 2 \pmod{12}$, *except possibly when* $g = 13$, *or* $g \equiv 7 \pmod{12}$, *or* $g = 17$ *and* $u \equiv 2 \pmod{24}$, *or* $(g, u) \in \{(15, 14), (21, 14), (11, 38), (17, 38)\}$;
- $u \equiv 3 \pmod{12}$, *except possibly when* $g \in \{13, 17\}$, *or* $g \equiv 7, 10 \pmod{12}$, *or* $u \in \{27, 39, 51\}$ *and* $g \equiv 5 \pmod{6}$, *or* $u = 27$ *and* $g \equiv 4 \pmod{12}$;

Theorem continued...(1)

Theorem

- $u \equiv 5 \pmod{12}$, *except possibly when* $g = \{13, 26, 44\}$, *or* $u \in \{29, 41\}$ *and* $g \equiv 2, 8 \pmod{24}$, *or* $g \equiv 5 \pmod{6}$, *or* $g \equiv 14 \pmod{24}$, *or* $u \equiv 17 \pmod{24}$ *and* $g \equiv 20 \pmod{24}$, *or* $u \equiv 17 \pmod{24}$ *and* $g \equiv 2 \pmod{24}$;
- $u \equiv 6 \pmod{12}$, *except when* $(g, u) = (3, 6)$, *and possibly when* $u = 6$ *and* $g \equiv 3 \pmod{6}$, $g \geq 9$, *or* $g \equiv 5 \pmod{6}$, *or* $(g, u) \in \{(15, 18), (21, 18)\}$;
- $u \equiv 7 \pmod{12}$, *except when* $(g, u) \in \{(1, 7), (1, 19)\}$, *and possibly when* $g \in \{5, 7\}$;
- $u \equiv 8 \pmod{12}$, *except possibly when* $(g, u) \in \{(11, 32), (11, 44), (17, 32), (17, 44)\}$;
- $u \equiv 9 \pmod{12}$, *except when* $(g, u) = (1, 9)$, *and possibly when* $g = 13$, *or* $g \equiv 5, 7, 10, 11 \pmod{12}$, *or* $u = 9$ *and* $g \equiv 1 \pmod{12}$, $g \geq 13$, *or* $u = 21$ *and* $g \equiv 4 \pmod{12}$;

Theorem continued...(2)

Theorem

- $u \equiv 10 \pmod{12}$, *except when* $(g, u) = (1, 10)$, *and possibly when* $g \in \{5, 7\}$;
- $u \equiv 11 \pmod{12}$, *except possibly when* $g = 26$, *or* $u = 35$ *and* $g \equiv 2, 8 \pmod{24}$, *or* $g \equiv 5 \pmod{6}$, *or* $u \equiv 11 \pmod{24}$ *and* $g \equiv 14, 20 \pmod{24}$, *or* $u \equiv 23 \pmod{24}$ *and* $g \equiv 2 \pmod{6}$, $g \geq 8$, *or* $(g, u) = (2, 23)$.

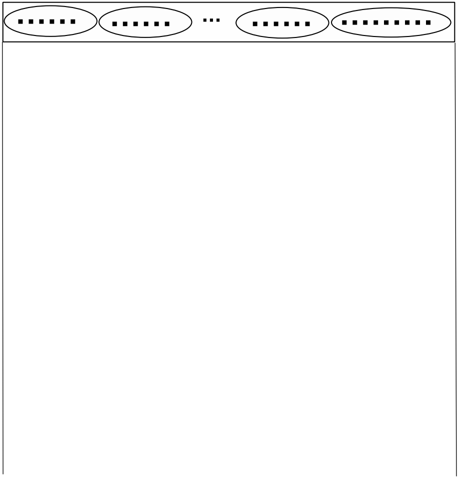
Close to optimal $4-GDCD$s

Theorem

- *When* $g \equiv 3 \pmod 6$, *there exists a close to optimal* $4-GDCD$ *of type* g^6 *with* $\left\lceil \frac{3g}{2} \left\lceil \frac{5g}{3} \right\rceil \right\rceil + 2$ *blocks, which is optimal when* $g = 3$.
- *When* $g \equiv 1 \pmod{12}$, $g \neq 13$, *there exists a close to optimal* $4-GDCD$ *of type* g^9 *having* $\left\lceil \frac{9g}{4} \left\lceil \frac{8g}{3} \right\rceil \right\rceil + 1$ *blocks, which is optimal when* $g = 1$.

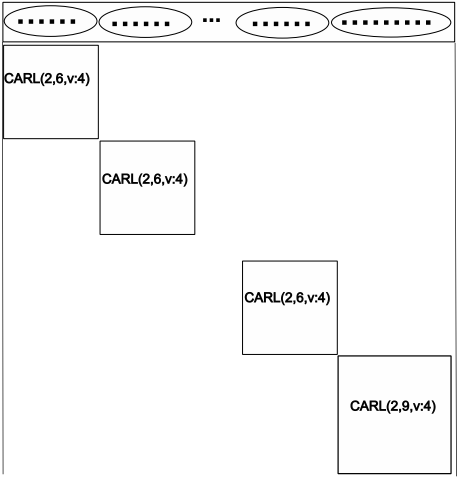
Wilson construction

Example: If $k \equiv 3 \pmod 6$, $k \geq 33$, there exists a $4-GDD$ of type $6^{\frac{k-9}{6}} 9^1$, $(V, \mathcal{G}, \mathcal{B})$.



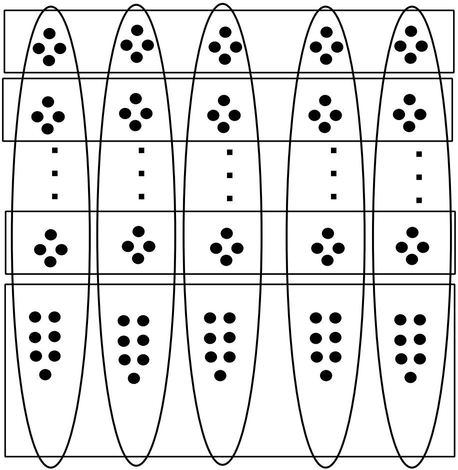
Wilson construction for *CARL*s with $w = 4$

Example: If $k \equiv 3 \pmod 6$, $k \geq 33$, there exists a $4 - GDD$ of type $6^{\frac{k-9}{6}} 9^1$, $(V, \mathcal{G}, \mathcal{B})$.



Construction of *CARL*s with smaller number of components

Example: $k = 5$ and $v \equiv 7 \pmod{12}$, $v \geq 31$:



References

Cohen, David M., Siddhartha R. Dalal, and Gardner C. Patton. 1997. *The AETG system: An approach to testing based on combinatorial design*, IEEE Transaction on Software Engineering **23**, no. 7, 437–444.

Gargano, L., J. Körner, and U. Vaccaro. 1993. *Sperner capacities*, Graphs and Combinatorics **9**, no. 1, 31–46.

Godbole, Anant P, Daphne E Skipper, and Rachel A Sunley. 1996. *t-covering arrays: upper bounds and poisson approximations*, Combinatorics, Probability and Computing **5**, no. 2, 105–117.

Katona, G. O. H. 1973. *Two applications (for search theory and truth functions) of sperner type theorems*, Periodica Mathematica Hungarica. Journal of the János Bolyai Mathematical Society **3**, 19–26. Collection of articles dedicated to the memory of Alfréd Rényi, II.

Kleitman, Daniel J. and Joel Spencer. 1973. *Families of k-independent sets*, Discrete Mathematics **6**, 255–262.

Maltais, Elizabeth and Lucia Moura. 2011. *Hardness results for covering arrays avoiding forbidden edges and error-locating arrays*, Theoretical Computer Science **412**, no. 46, 6517–6530.

Thank you!