Some Examples of Combinatorial Generation

Brendan McKay
Australian National University

and various colleagues to be mentioned...

In honour of Gordon's $(61+\delta)$-th birthday

The Australian National University

GPO Box 4, Canberra, ACT 2601
Telegrams & cables NATUNIV Canberra
Telephone 062-49 5111
Telex AA 62760 NATUNI

reference

Dear Gordon,

Here it is.

Tape details:

Density: 1600
Labels: NONE
Code: ASCII
Records: Fixed length 80 bytes, blank fill
Blocks: Fixed length 800 bytes = 10 records

Files:

#	records	contents	
1	110	nauty.h	} compile & load all these together to run dreadnaut
2	935	nautil.c	
3	748	naugraph.c	
4	897	dreadnaut.c	
5	50	test.dat	— sample input for dreadnaut
6	1322	cayleygraphs	— Cayley graphs (as converged by Cleyn
7	1020	transgraphs	— transitive graphs to 18 points
8-14	copies of files 1-7		

Note that dreadnaut is only suitable for small tasks — you need to call nauty() yourself for anything serious.

Please keep a log of all the problems you have running these programs, even the most trivial. I'm trying to make them as portable as possible. Good luck.

Regards,
Brendan

CONSTRUCTING THE CUBIC GRAPHS ON UP TO 20 VERTICES

BY

BRENDAN D. MCKAY
and
GORDON F. ROYLE

February 1985

Brendan D. McKay
Computer Science Department
Australian National University
Canberra
AUSTRALIA

Gordon F. Royle
Department of Mathematics
University of Western Australia
Nedlands
Western Australia 6009

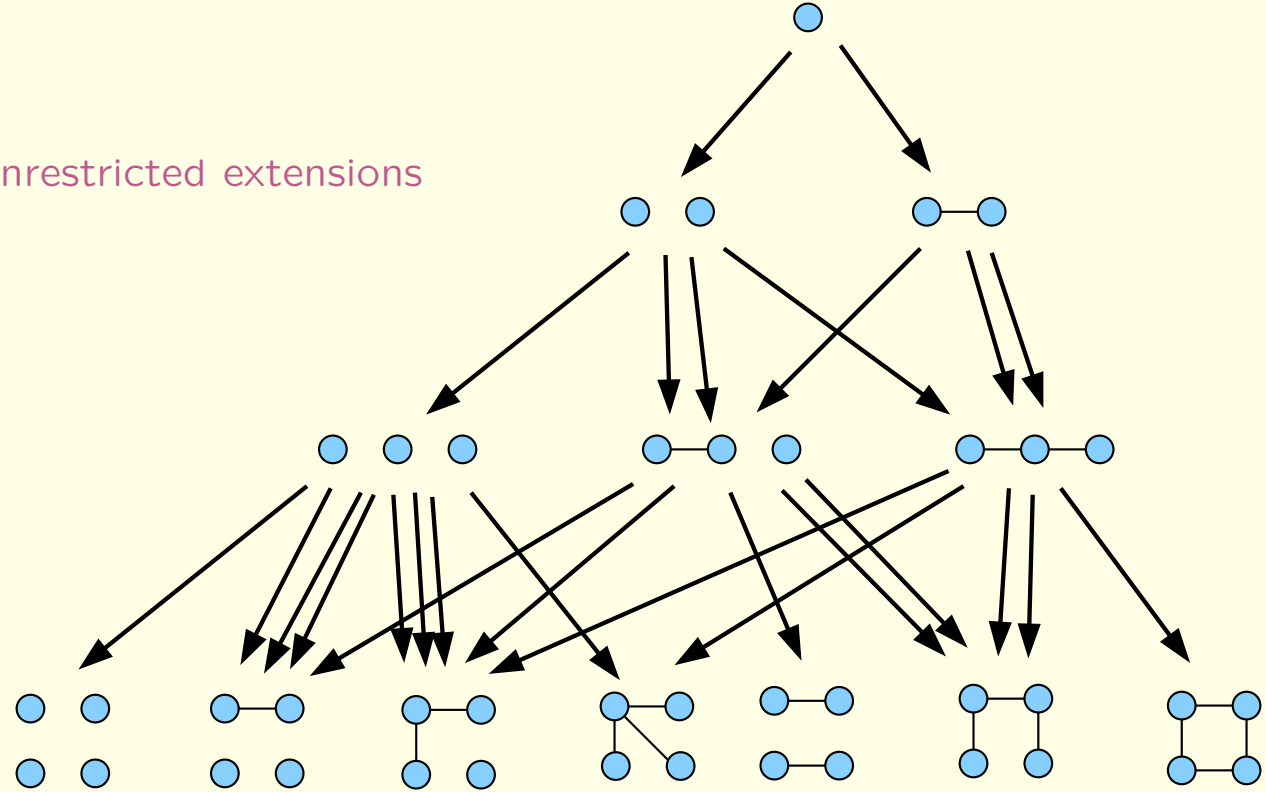
Canonical construction path method

Example of triangle-free graphs.

Canonical construction path method

Example of triangle-free graphs.

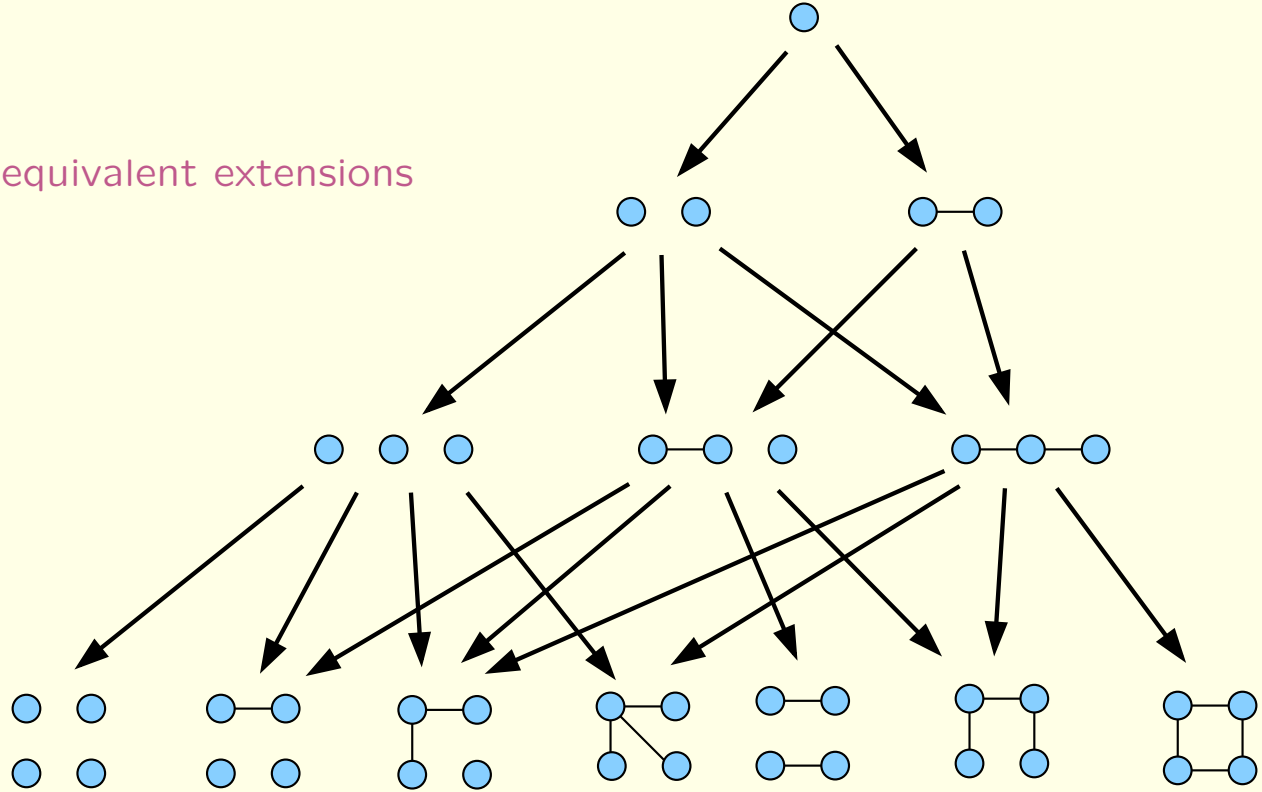
Unrestricted extensions



Canonical construction path method

Example of triangle-free graphs.

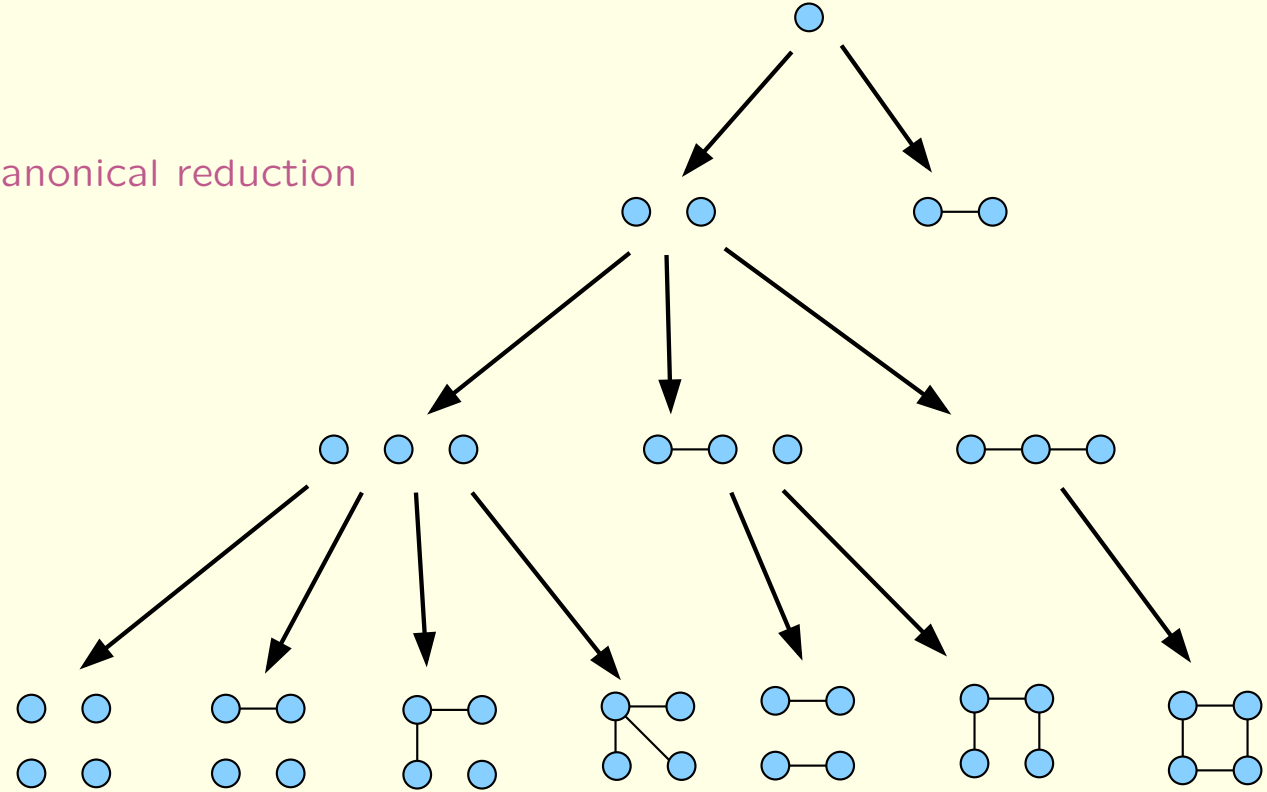
Inequivalent extensions



Canonical construction path method

Example of triangle-free graphs.

Canonical reduction



Extremal graphs

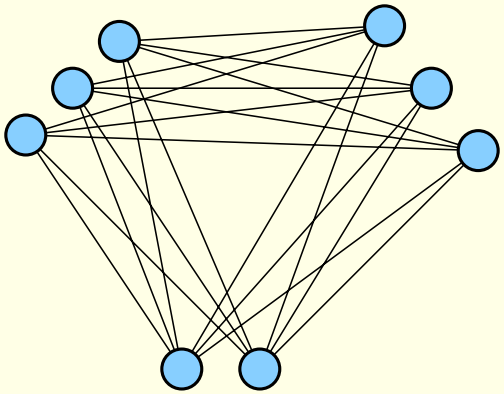
Coauthor: Narjess Afzaly

In 1941, Turán proved that the most edges an n-vertex graph without K_r can have is when it is a complete $(r-1)$-partite graph with near-equal sides.

Extremal graphs

Coauthor: Narjess Afzaly

In 1941, Turán proved that the most edges an n-vertex graph without K_r can have is when it is a complete $(r-1)$-partite graph with near-equal sides.

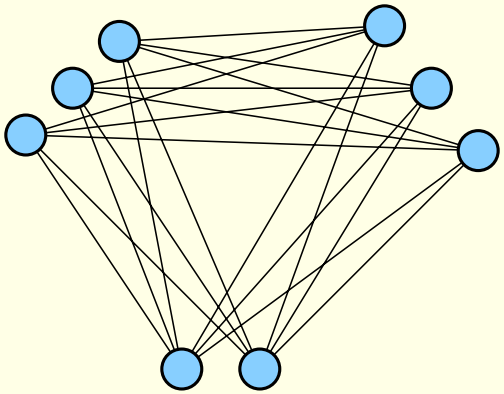


$K_{2,3,3}$, extreme for K_4

Extremal graphs

Coauthor: Narjess Afzaly

In 1941, Turán proved that the most edges an n-vertex graph without K_r can have is when it is a complete $(r-1)$-partite graph with near-equal sides.



$K_{2,3,3}$, extreme for K_4

In general, if $\mathcal{H}$ is a collection of graphs,

$$ex(n, \mathcal{H}) = \begin{cases} \text{the greatest number of edges that a graph on} \\ n \text{ vertices can have without having a member} \\ \text{of } \mathcal{H} \text{ as a subgraph.} \end{cases}$$

Extremal graphs (continued)

We will make graphs by adding one vertex at a time.

Usually, for large orders there are extremely many $\mathcal{H}$-free graphs but relatively few with close to $ex(n, \mathcal{H})$ edges.

Extremal graphs (continued)

We will make graphs by adding one vertex at a time.

Usually, for large orders there are extremely many $\mathcal{H}$-free graphs but relatively few with close to $ex(n, \mathcal{H})$ edges.

For example, graphs on 25 vertices with no C_4 or C_5.

edges	graphs
...	
45	30651877057
46	895164804
47	15409643
48	176966
49	1799
50	17

Extremal graphs (continued)

If we remove a vertex from an extremal graph, the resulting graph might not be extremal.

So in order to generate the extremal graphs by adding one vertex at a time, it is necessary to also generate non-extremal graphs.

The definition of "canonical reduction" (i.e., which vertex is removed to make the parent) is designed so that the ancestors of an extremal graph are reasonably close to extremal.

Extremal graphs (continued)

If we remove a vertex from an extremal graph, the resulting graph might not be extremal.

So in order to generate the extremal graphs by adding one vertex at a time, it is necessary to also generate non-extremal graphs.

The definition of "canonical reduction" (i.e., which vertex is removed to make the parent) is designed so that the ancestors of an extremal graph are reasonably close to extremal.

This is done by a sequence of rules:

1. Choose a vertex of minimum degree.
2. If there is more than one vertex of minimum degree, choose one that is adjacent to the most other vertices of minimum degree.
3. Etc.
4. If no vertex is chosen by now, use `nauty` to choose one.

Extremal graphs (continued)

Graphs are classified into families $\mathcal{G}_{\mathcal{H}}(n, e, d, m, t; d', m', t')$, where

- $n =$ the number of vertices
- $e =$ the number of edges
- $d =$ the minimum degree
- $m =$ the number of vertices of minimum degree
- $t =$ whether there are any adjacent vertices of minimum degree
- $n-1, e-d, d', m', t' =$ those same parameters for the parent

Extremal graphs (continued)

Then a lot of lemmas are applied to identify possible parameters for the classes of the parent. For example, for $\mathcal{H} = \{C_4, C_5\}$, the parents of family $\mathcal{G}_{\mathcal{H}}(26, 52, 3, 3, F; 3, 3, F)$ are calculated to lie in one of

$\mathcal{G}_{\mathcal{H}}(25, 49, 3, 3, F; 3, 5, T)$,
$\mathcal{G}_{\mathcal{H}}(25, 49, 3, 3, F; 3, 4, T)$,
$\mathcal{G}_{\mathcal{H}}(25, 49, 3, 3, F; 3, 5, F)$, and
$\mathcal{G}_{\mathcal{H}}(25, 49, 3, 3, F; 3, 4, F)$.

Extremal graphs (continued)

Since the computation is very long-running and families
$\mathcal{G}_{\mathcal{H}}(n, e, d, m, t; d', m', t')$
appear as ancestors for many different output sizes, we store each such family on disk using a custom compression method that usually needs only 1–2 bytes per graph.

Implementation

There is a controller program that decides which families are potentially needed as ancestors for a required output size, then makes the missing files using a multi-threaded worker process.

$ex(n; \{C_4, C_5\})$

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	3	4	6	7	9	10	12
10	14	16	18	21	23	25	28	30	33	35
20	38	42	43	45	48	50	53	55	58	62
30	65	67	70	73	77	79	82	86	89	93
40	96	100	105	107	110					

Typical construction path for an extremal graph with 43 vertices:

Key: vertices,edges, green means extremal.

1,0, 2,1, 3,2, 4,3, 5,4, 6,6, 7,7, 8,9, 9,10, 10,12, 11,14, 12,16, 13,18, 14,20, 15,22, 16,24, 17,26, 18,28, 19,30, 20,33, 21,35, 22,38, 23,40, 24,43, 25,46, 26,49, 27,52, 28,56, 29,58, 30,61, 31,64, 32,67, 33,70, 34,74, 35,77, 36,81, 37,84, 38,88, 39,92, 40,96, 41,100, 42,105, **43,107**

$ex(n; \{C_4\})$

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	3	4	6	7	9	11	13
10	16	18	21	24	27	30	33	36	39	42
20	46	50	52	56	59	63	67	71	76	80
30	85	90	92	96	102	106	110	113	117	122
40	127									

$ex(n; \{C_3, C_4\})$

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	2	3	5	6	8	10	12
10	15	16	18	21	23	26	28	31	34	38
20	41	44	47	50	54	57	61	65	68	72
30	76	80	85	87	90	95	99	104	109	114
40	120	124	129	134	139	145	150	156	162	168
50	175	176	178							

$ex(n; \{C_3, C_4, C_5\})$

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	2	3	4	6	7	9	10
10	12	14	16	18	21	22	24	26	29	31
20	34	36	39	42	45	48	52	53	56	58
30	61	64	67	70	74	77	81	84	88	92
40	96	100	105	106	108	110	115	118	122	126
50	130	134	138	142	147	151	156	160	165	170
60	175	180	186	187	189					

$ex(n; \{C_4, C_{\text{odd}}\})$ (Zarankiewicz problem)

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	2	3	4	6	7	9	10
10	12	14	16	18	21	22	24	26	29	31
20	34	36	39	42	45	48	52	53	56	58
30	61	64	67	70	74	77	81	84	88	92
40	96	100	105	106	108	110	115	118	122	126
50	130	134	138	142	147	151	156	160	165	170
60	175	180	186	187	189					

$ex(n; \{C_4, C_{\text{odd}}\})$ (Zarankiewicz problem)

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	2	3	4	6	7	9	10
10	12	14	16	18	21	22	24	26	29	31
20	34	36	39	42	45	48	52	53	56	58
30	61	64	67	70	74	77	81	84	88	92
40	96	100	105	106	108	110	115	118	122	126
50	130	134	138	142	147	151	156	160	165	170
60	175	180	186	187	189					

Question: Is $ex(n; \{C_4, C_{\text{odd}}\}) = ex(n; \{C_3, C_4, C_5\})$ for all n?

Block designs

Coauthors: Daniel Heinlein, Andrei Ivanov, Patric Östergård

For integers v, k, λ, a 2-(v, k, λ) design is a collection of k-subsets ("blocks") of a set of v "points", such that every pair of points lie in exactly λ blocks.

Block designs

Coauthors: Daniel Heinlein, Andrei Ivanov, Patric Östergård

For integers v, k, λ, a 2-(v, k, λ) design is a collection of k-subsets ("blocks") of a set of v "points", such that every pair of points lie in exactly λ blocks.

For example,

$$\{\{0,2,3\},\{1,2,3\},\{0,1,4\},\{1,2,4\},\{0,3,4\},$$
$$\{0,1,5\},\{0,2,5\},\{1,3,5\},\{2,4,5\},\{3,4,5\}\}$$

is a 2-$(6,3,2)$ design. Implied parameters are

$b =$ the number of blocks $= 10$

$r =$ the number of blocks containing each point $= 5$.

Block designs

Coauthors: Daniel Heinlein, Andrei Ivanov, Patric Östergård

For integers v, k, λ, a 2-(v, k, λ) design is a collection of k-subsets ("blocks") of a set of v "points", such that every pair of points lie in exactly λ blocks.

For example,

$$\{\{0,2,3\},\{1,2,3\},\{0,1,4\},\{1,2,4\},\{0,3,4\},$$
$$\{0,1,5\},\{0,2,5\},\{1,3,5\},\{2,4,5\},\{3,4,5\}\}$$

is a 2-$(6,3,2)$ design. Implied parameters are

$b =$ the number of blocks $= 10$

$r =$ the number of blocks containing each point $= 5$.

$$bk = vr, \qquad b\binom{k}{2} = \lambda\binom{v}{2}, \qquad b \geq v.$$

Incidence matrix

Consider our 2-(6, 3, 2) design:

$$\{\{0,2,3\},\{1,2,3\},\{0,1,4\},\{1,2,4\},\{0,3,4\},$$
$$\{0,1,5\},\{0,2,5\},\{1,3,5\},\{2,4,5\},\{3,4,5\}\}$$

Incidence matrix

Consider our 2-(6, 3, 2) design:

$$\{\{0,2,3\},\{1,2,3\},\{0,1,4\},\{1,2,4\},\{0,3,4\},$$
$$\{0,1,5\},\{0,2,5\},\{1,3,5\},\{2,4,5\},\{3,4,5\}\}$$

The incidence matrix is

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Each point (row) is a vector in $\{0,1\}^b$, all row (point) sums are r, all column (block) sums are k, and all row inner products are λ.

The project

Two designs D_1, D_2 are isomorphic if there is a bijection from the points of D_1 to the points of D_2 which maps the multiset of blocks of D_1 to the multiset of blocks of D_2.

An automorphism of a design is a permutation of the points which preserves the multiset of blocks; i.e. an isomorphism of the design to itself.

The project

Two designs D_1, D_2 are isomorphic if there is a bijection from the points of D_1 to the points of D_2 which maps the multiset of blocks of D_1 to the multiset of blocks of D_2.

An automorphism of a design is a permutation of the points which preserves the multiset of blocks; i.e. an isomorphism of the design to itself.

The aim of the project is to compile complete lists of nonisomorphic 2-(v, k, λ) designs for as many parameter sets as possible, and make them available on the internet.

Almost all computations are done in duplicate, to enhance precision.

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{pmatrix}$$

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{pmatrix}$$

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{pmatrix}$$

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ & & & & & & & & & \end{pmatrix}$$

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Construction methods

Both methods worked by adding one row at a time, reducing the partial designs by isomorphism class.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Pruning of the search took into account additional information that could be calculated, such as the possible amounts by which two blocks can intersect. Also, when most of the rows are present, sometimes it is possible to tell that completion is impossible.

Ivanov applied the orderly method, whereby the partial designs are kept in a unique extremal form such that removing the last row from a extremal form gives the extremal form of the smaller partial design.

Ivanov applied the orderly method, whereby the partial designs are kept in a unique extremal form such that removing the last row from a extremal form gives the extremal form of the smaller partial design.

McKay applied the canonical construction path method that gives each partial design a unique row such that removing the row gives the parent.

Ivanov applied the orderly method, whereby the partial designs are kept in a unique extremal form such that removing the last row from a extremal form gives the extremal form of the smaller partial design.

McKay applied the canonical construction path method that gives each partial design a unique row such that removing the row gives the parent.

In each case, the definitions are chosen so that many partial designs can be seen to never lead to a completed design.

Ivanov applied the orderly method, whereby the partial designs are kept in a unique extremal form such that removing the last row from a extremal form gives the extremal form of the smaller partial design.

McKay applied the canonical construction path method that gives each partial design a unique row such that removing the row gives the parent.

In each case, the definitions are chosen so that many partial designs can be seen to never lead to a completed design.

Determining the possibilities for the next row requires solving a set of integer inequalities. Information about the solutions for each row can be used to speed up the computation for additional rows.

Example: $v = 9,\ k = 3,\ \lambda = 4,\ r = 16,\ b = 48$

$\lvert Aut(D)\rvert$	designs	simple	trans	simtrans
1	16534655	275	0	0
2	47286	37	0	0
3	1127	6	0	0
4	1450	4	0	0
6	221	5	0	0
8	171	0	0	0
9	8	1	8	1
12	38	0	0	0
16	26	1	0	0
18	10	1	10	1
24	14	1	0	0
32	8	0	0	0
40	1	0	0	0
48	6	0	0	0
54	3	1	3	1
72	1	0	1	0
80	1	0	0	0
108	2	0	2	0
384	1	0	0	0
432	1	0	1	0
2880	1	0	0	0

16585031 designs generated (332 simple); 3476.64 sec

Results

https://zenodo.org/records/8303393

100 parameter sets

214 GB compressed

Results

https://zenodo.org/records/8303393

100 parameter sets

214 GB compressed

We only provide complete compilations, though there is a program that can make them at "random".

Results

https://zenodo.org/records/8303393

100 parameter sets

214 GB compressed

We only provide complete compilations, though there is a program that can make them at "random".

Often the number of designs becomes too large with only a modest increase in the parameters:

2-(8,3,12) : about 4×10^{12} designs
2-(9,4,9) : about 2×10^{15} designs
2-(15,3,2) : about 1.5×10^{21} designs