Southwest State University
Department of computer sciences

SEARCHING FOR ORTHOGONAL LATIN SQUARES VIA CELLS MAPPING AND BOINC-BASED CUBE-AND-CONQUER

Vatutin E.I., Zaikin O.S., Manzuk M.O., Nikitina N.N.

Moscow, 2021

What is Latin squares?

$$A = \left\| a_{ij} \right\|$$

$$i,\ j = \overline{1,\ N}$$

$$N = |S|$$

$$S = \{0, 1, 2, \ldots, N-1\}$$

$$\forall i,\ j,\ k = \overline{1,\ N},\ j \neq k : (a_{ij} \neq a_{ik}) \wedge (a_{ji} \neq a_{ki})$$

$$\forall i,\ j = \overline{1,\ N},\ i \neq j : (a_{ii} \neq a_{jj}) \wedge (a_{N-i+1,\ N-i+1} \neq a_{N-j+1,\ N-j+1})$$

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

Normalized LS of order 10

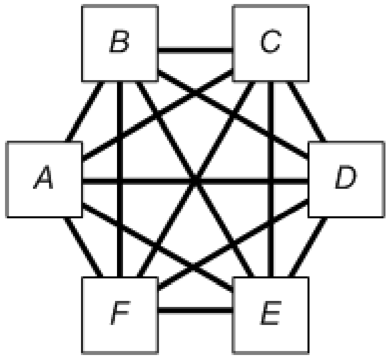
$$N! \times (N-1)!$$

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

Normalized DLS of order 10

$$(N-1)!$$

Why is this interesting?



Applied problems:

- experiment planning
- cryptography
- error correcting codes
- scheduling
- algebra, combinatorics, statistics, …

Mathematical problems:

- **existence of a triple of MOLS / MODLS of order 10 (or larger clique)**
- increasing world record of orthogonality characteristic for pseudo triple of MOLS (291/300) or MODLS (274/300)
- generating functions
- asymptotic behavior of combinatorial characteristics based on DLSs (OEIS)
- number theory (relations between different fields of knowledge)
- magic squares
- Sudoku (LS of order 9 with additional constraints)

Brief history and approaches

- Euler — triple of MOLS of order 10 does not exist (**disproved**);
- Parker et al. (1960) — **first pair of MOLS** of order 10 using transversals;
- Brown et al. (1995) — horyzontally symmetric DLS, **1:4**;
- Zaikin et al. (2015—2016) — **SAT approach (system of Boolean equations)**, ~100 CFs of ODLS, **onces**;
- Vatutin et al. (from 2017) — **RS+LBF generator -> transversals + DLX**, ~1M CFs of ODLS, **twices**, **1:3**;
- Vatutin et al. (2017) — **plane symmetry generator -> transversals + DLX**, ~200k CFs of ODLS, **1:6**, **1:8**, **rhombus-4**, **line-4**, **line-5**, **loop-4**, **fish**;
- Vatutin et al. (from 2017) — **central symmetry, partial central and plane symmetries, generalized symmetries, neighborhoods of generalized symmetris generator -> canonizer -> postprocessor**, ~17M CFs of ODLS, **1:5**, **1:7**, **1:10**, **rhombus-3**, **cross**, **flyer**, **tree-1**, **Venus**, **Daedalus-8**, **Daedalus-10**, **robot**, **stingray**;
- Vatutin et al. (from 2019) — **transversal free (CMS based) search for ODLS.**

Classical search for pairs of ODLS of order 10: Euler-Parker method

L. Euler expected that for N=10 ODLS doesn't exist

First pair — Parker et al., 1960

0	1	2	3	4	5	6	7	8	9
1	2	0	4	3	7	9	8	5	6
7	3	5	9	0	4	8	6	2	1
3	5	6	8	9	0	4	1	7	2
4	9	7	2	6	8	1	5	0	3
5	8	4	6	7	1	3	2	9	0
8	4	9	1	2	3	7	0	6	5
6	7	3	0	1	2	5	9	4	8
9	0	1	5	8	6	2	4	3	7
2	6	8	7	5	9	0	3	1	4

0	1	2	3	4	5	6	7	8	9
7	5	1	9	2	8	0	4	6	3
1	0	3	4	6	7	5	2	9	8
9	8	4	7	5	2	1	0	3	6
6	7	9	0	8	3	2	1	5	4
4	6	5	1	0	9	8	3	2	7
2	3	8	5	1	6	4	9	7	0
5	2	7	8	3	4	9	6	0	1
3	4	6	2	9	0	7	8	1	5
8	9	0	6	7	1	3	5	4	2

SAT@Home, 04.2015

0	1	2	3	4	5	6	7	8	9
4	9	0	8	5	6	3	1	2	7
2	5	7	9	6	4	0	8	1	3
9	0	4	6	8	7	1	5	3	2
6	7	5	2	1	3	8	0	9	4
1	8	3	5	7	2	9	6	4	0
7	3	1	0	9	8	4	2	6	5
8	2	6	4	0	9	5	3	7	1
3	4	8	1	2	0	7	9	5	6
5	6	9	7	3	1	2	4	0	8

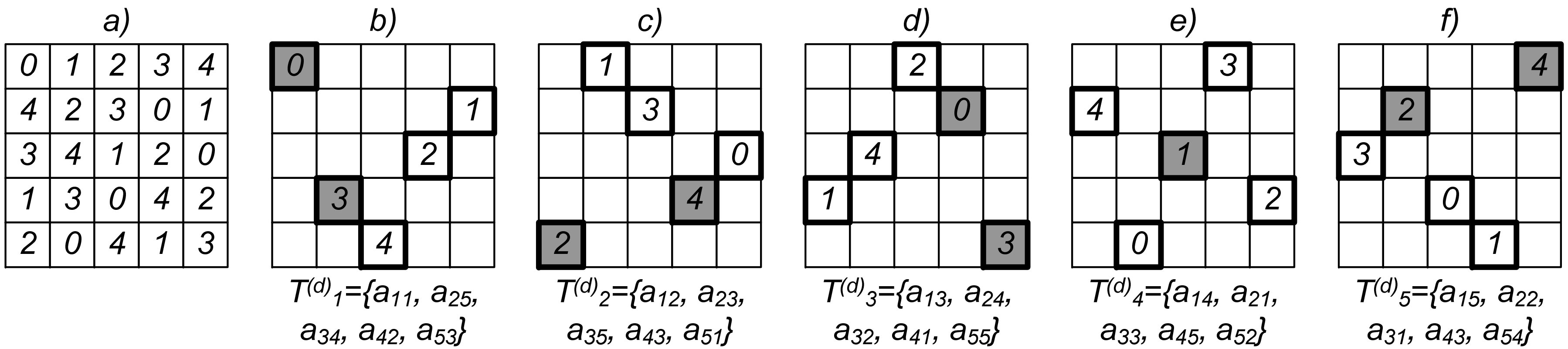
0	1	2	3	4	5	6	7	8	9
6	5	9	7	0	8	2	3	1	4
4	7	1	2	3	9	8	0	6	5
1	2	0	4	5	3	7	6	9	8
2	6	8	0	9	4	1	5	3	7
8	4	6	9	2	7	0	1	5	3
5	0	4	6	8	2	3	9	7	1
9	3	5	1	7	6	4	8	0	2
7	8	3	5	6	1	9	4	2	0
3	9	7	8	1	0	5	2	4	6

Gerasim@Home, 04.2017

Very rare combinatorial objects:
~30 millions DLS of order 10
has only **1 pair of ODLS**!

Searching for ODLS: approaches

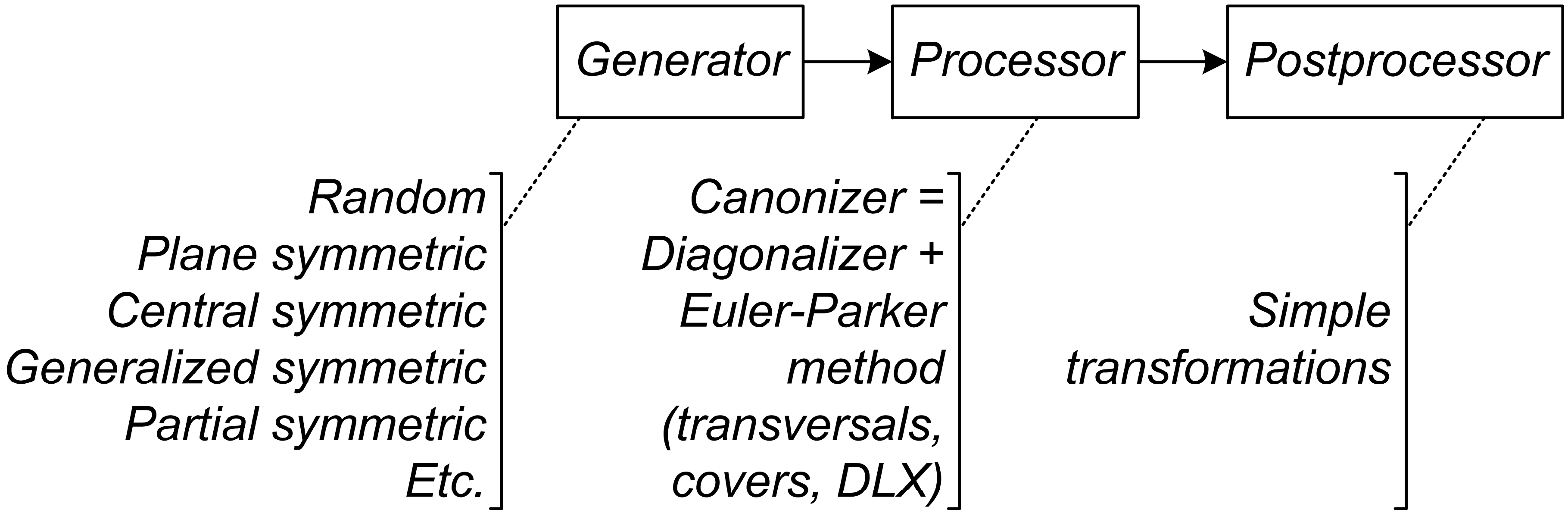
- Brute Force + backtracking + clippings + ordering + … (very long)
- SAT (some tens of hours, long)
- filling by pairs of elements [a_{ij}, b_{ij}] (long)
- using transversals (fast) – **200 – 800 DLS/s** for different algorithms!
- using transversals with canonizer (**~8000 DLS/s** effective pace)



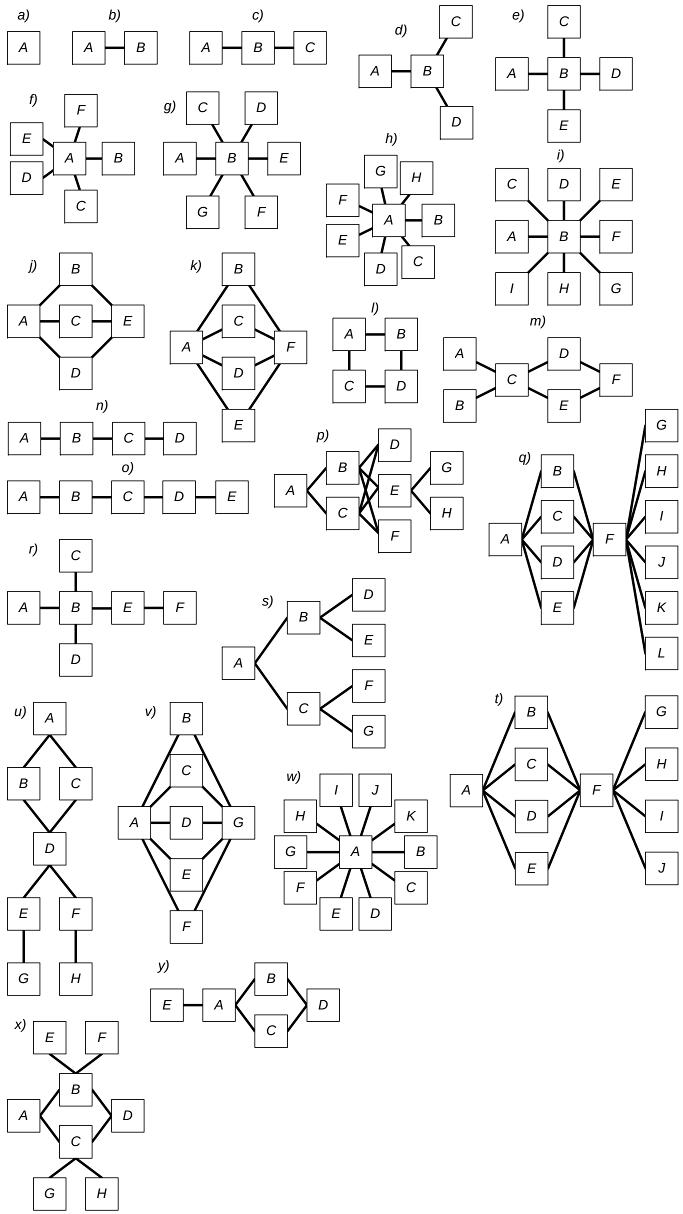
- DLS generators: **~6 600 000 DLS/s**
- bottleneck: Euler-Parker method based on transversals

Aim 1: increasing pace of ODLS generating

Can we mix high pace of DLS generators (**6 600 000 DLS/s**) and low pace of Euler-Parker method (**8 000 DLS/s** using diagonalizing with ODLS check)?



Aim 2: searching for new rare combinatorial structures



Aim 3: searching for pseudo triples of MODLS with high orthogonality characteristic (OC)

OC=74 (DLS A ↔ DLS C)

OC=100 (DLS A ↔ DLS B)

OC=100 (DLS B ↔ DLS C)

DLS A

0	1	2	3	4	5	6	7	8	9
5	8	6	7	9	4	0	3	1	2
6	4	7	8	2	9	1	0	5	3
7	6	5	4	8	1	2	9	3	0
3	0	9	2	1	8	7	5	4	6
2	5	0	1	6	3	8	4	9	7
8	3	4	5	7	0	9	2	6	1
1	2	3	9	5	7	4	6	0	8
9	7	8	0	3	6	5	1	2	4
4	9	1	6	0	2	3	8	7	5

DLS B

0	1	2	3	4	5	6	7	8	9
1	2	3	4	0	9	5	6	7	8
4	0	1	7	3	6	2	8	9	5
8	7	6	5	9	0	4	3	2	1
7	6	5	0	8	1	9	4	3	2
9	8	7	6	5	4	3	2	1	0
5	9	8	2	6	3	7	1	0	4
3	5	0	8	7	2	1	9	4	6
2	3	4	9	1	8	0	5	6	7
6	4	9	1	2	7	8	0	5	3

DLS C

0	1	2	3	4	5	6	7	8	9
5	9	0	2	3	6	7	8	4	1
1	7	3	5	9	0	8	6	2	4
9	2	1	6	0	4	3	5	7	8
3	4	9	8	2	7	5	0	1	6
7	5	6	9	1	8	4	3	0	2
8	3	4	0	7	2	1	9	6	5
6	0	5	7	8	1	2	4	9	3
4	8	7	1	6	3	9	2	5	0
2	6	8	4	5	9	0	1	3	7

ODLS pair BC

00	11	22	33	44	55	66	77	88	99
15	29	30	42	03	96	57	68	74	81
41	07	13	75	39	60	28	86	92	54
89	72	61	56	90	04	43	35	27	18
73	64	59	08	82	17	95	40	31	26
97	85	76	69	51	48	34	23	10	02
58	93	84	20	67	32	71	19	06	45
36	50	05	87	78	21	12	94	49	63
24	38	47	91	16	83	09	52	65	70
62	46	98	14	25	79	80	01	53	37

ODLS pair AB

00	11	22	33	44	55	66	77	88	99
51	82	63	74	90	49	05	36	17	28
64	40	71	87	23	96	12	08	59	35
78	67	56	45	89	10	24	93	32	01
37	06	95	20	18	81	79	54	43	62
29	58	07	16	65	34	83	42	91	70
85	39	48	52	76	03	97	21	60	14
13	25	30	98	57	72	41	69	04	86
92	73	84	09	31	68	50	15	26	47
46	94	19	61	02	27	38	80	75	53

ODLS pair AC

00	11	22	33	44	55	66	77	88	99
—	89	60	72	93	46	07	38	14	21
61	47	73	85	29	90	18	06	52	34
79	62	51	—	80	—	23	95	37	08
—	04	—	28	12	87	75	50	41	—
27	—	—	19	—	—	84	43	—	—
—	—	—	—	—	02	91	—	—	15
16	20	35	97	58	71	42	64	09	83
94	78	—	01	36	63	59	—	25	40
—	96	—	—	05	—	30	81	—	57

Transversals free search for ODLS: SODLS

SODLS:
A329685, A287761, A287762

ESODLS:
A309210, A309598, A309599

DSODLS:
A333366, A333367, A333671

0	3	9	5	8	7	2	6	1	4
1	6	7	4	5	3	9	8	2	0
2	0	3	7	9	4	1	5	8	6
3	2	8	9	1	6	0	4	5	7
4	1	0	6	7	8	5	3	9	2
5	4	1	8	3	2	7	0	6	9
6	5	2	1	4	9	8	7	0	3
7	9	6	0	2	5	4	1	3	8
8	7	5	2	6	0	3	9	4	1
9	8	4	3	0	1	6	2	7	5

0	1	2	3	4	5	6	7	8	9
3	6	0	2	1	4	5	9	7	8
9	7	3	8	0	1	2	6	5	4
5	4	7	9	6	8	1	0	2	3
8	5	9	1	7	3	4	2	6	0
7	3	4	6	8	2	9	5	0	1
2	9	1	0	5	7	8	4	3	6
6	8	5	4	3	0	7	1	9	2
1	2	8	5	9	6	0	3	4	7
4	0	6	7	2	9	3	8	1	5

SODLS: $B = A^T$

- Search without transversals using with high pace
- DSODLS is a subset of SODLS
- SODLS can be extended to ESODLS (Ed's SODLS)

Bennett F.E., Beiliang Du, Hantao Zhang. Existence of self-orthogonal diagonal Latin squares with a missing subsquare // Discrete Mathematics. Vol. 261. 2003. pp. 69-86.

SODLS can be extended for ESODLS

ESODLS (Ed's SODLS) — MODLS from DLSs within same main class

OEIS sequences (SODLS, H. White):

- A287761 — **1, 0, 0, 2, 4, 0, 64, 1152, 224832;**
- A287762 — **1, 0, 0, 48, 480, 0, 322560, 46448640, 81587036160.**

OEIS sequences (ESODLS, new):

- A309210 — **1, 0, 0, 1, 1, 0, 5, 23;**
- A309598 — **1, 0, 0, 2, 4, 0, 256, 4608;**
- A309599 — **1, 0, 0, 48, 480, 0, 1290240, 185794560.**

This site is supported by donations to The OEIS Foundation.

THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES ®

founded in 1964 by N. J. A. Sloane

Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

A309598 Number of extended self-orthogonal diagonal Latin squares of order n with ordered first string. 0

1, 0, 0, 2, 4, 0, 256, 4608 (list; graph; refs; listen; history; text; internal format)

OFFSET 1,4

COMMENTS A self-orthogonal diagonal Latin square (SODLS) is a diagonal Latin square orthogonal to its transpose. An extended self-orthogonal diagonal Latin square (ESODLS) is a diagonal Latin square that has an orthogonal diagonal Latin square from the same main class. SODLS is a special case of ESODLS.

LINKS Table of n, a(n) for n=1..8.
E. I. Vatutin, Discussion about properties of diagonal Latin squares (in Russian)
Index entries for sequences related to Latin squares and rectangles

EXAMPLE
```
The diagonal Latin square
  0 1 2 3 4 5 6 7 8 9
  1 2 0 4 5 7 9 8 6 3
  5 0 1 6 3 9 8 2 4 7
  9 3 5 8 2 1 7 4 0 6
  4 6 3 5 7 8 0 9 2 1
  8 4 6 9 1 3 2 5 7 0
  7 8 9 0 6 4 5 1 3 2
  2 9 4 7 8 0 3 6 1 5
  6 5 7 1 0 2 4 3 9 8
  3 7 8 2 9 6 1 0 5 4
has orthogonal diagonal Latin square
  0 1 2 3 4 5 6 7 8 9
  3 5 9 8 6 2 0 1 4 7
  4 3 8 7 2 1 9 0 5 6
  6 9 3 4 8 0 1 2 7 5
  7 2 0 1 9 3 5 8 6 4
  2 0 1 5 7 6 4 9 3 8
  8 6 4 2 0 9 7 5 1 3
  1 7 6 0 5 4 8 3 9 2
  9 8 5 6 1 7 3 4 2 0
  5 4 7 9 3 8 2 6 0 1
from the same main class.
```

CROSSREFS Cf. A287761.
Sequence in context: A287761 A009512 A317411 * A305570 A287651 A163259
Adjacent sequences: A309595 A309596 A309597 * A309599 A309600 A309601

KEYWORD nonn,more

AUTHOR Eduard I. Vatutin, Aug 09 2019

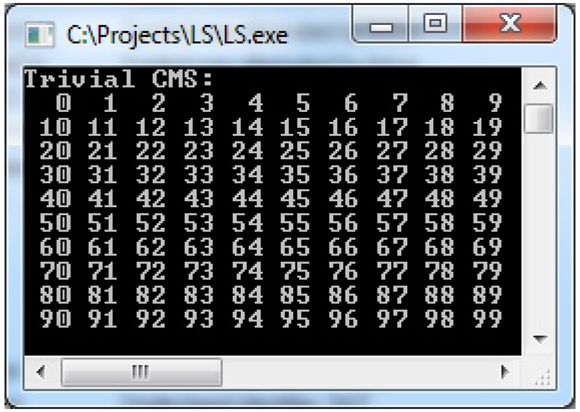
STATUS approved

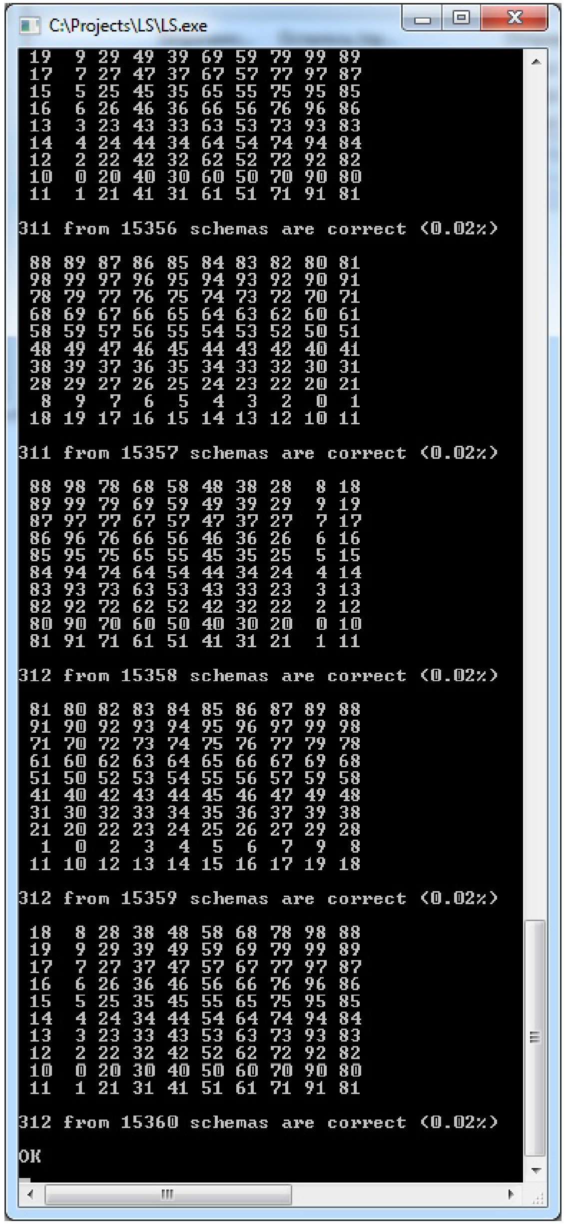
How we can find ESODLS? CMS-based search…

Cells Mapping Schema (CMS) — bijective mapping for N^2 cells of square with some special properties

SODLS — only one of them…

M-transformations
(15 360 combinations for N=10)





CMS example

DLS A

0	1	2	3	4	5	6	7	8	9
1	2	0	4	3	7	9	8	6	5
7	6	1	5	9	3	0	2	4	8
5	0	8	7	6	2	4	3	9	1
6	9	5	2	8	1	3	4	0	7
3	4	7	1	5	9	8	0	2	6
2	8	4	0	7	6	5	9	1	3
9	5	3	8	1	4	2	6	7	0
4	7	9	6	0	8	1	5	3	2
8	3	6	9	2	0	7	1	5	4

CMS

77	17	97	67	57	47	37	7	87	27
71	11	91	61	51	41	31	1	81	21
79	19	99	69	59	49	39	9	89	29
76	16	96	66	56	46	36	6	86	26
75	15	95	65	55	45	35	5	85	25
74	14	94	64	54	44	34	4	84	24
73	13	93	63	53	43	33	3	83	23
70	10	90	60	50	40	30	0	80	20
78	18	98	68	58	48	38	8	88	28
72	12	92	62	52	42	32	2	82	22

DLS B

6	8	1	9	0	4	3	7	5	2
5	2	3	8	4	9	0	1	7	6
0	5	4	3	6	7	1	9	2	8
2	9	7	5	8	3	4	6	1	0
4	7	0	6	9	1	2	5	8	3
1	3	2	7	5	8	6	4	0	9
8	4	9	0	1	2	7	3	6	5
9	1	8	2	3	6	5	0	4	7
7	6	5	1	2	0	9	8	3	4
3	0	6	4	7	5	8	2	9	1

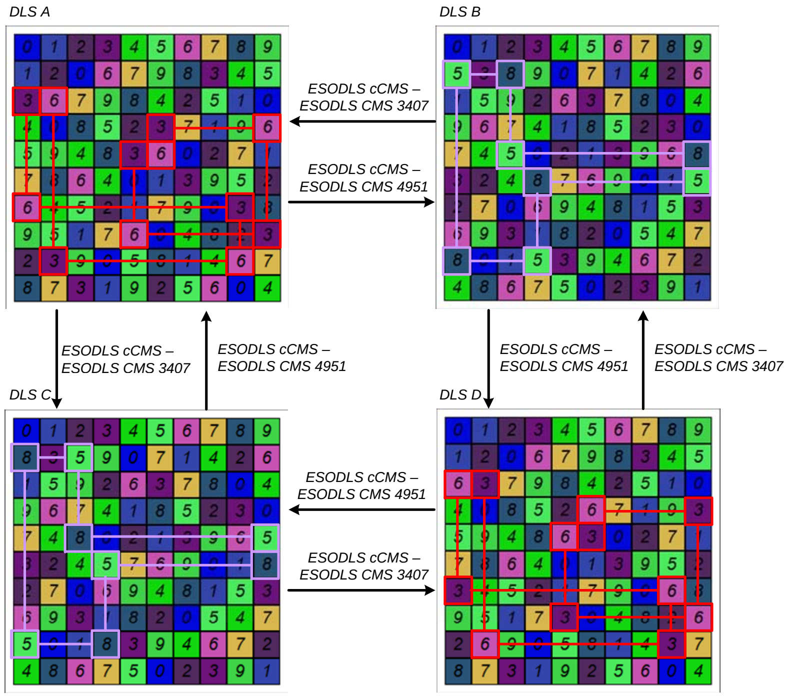
Loops structure for CMS

- CMS cells CMS[i1] -> CMS[i2] -> … -> CMS[iM] -> CMS[i1] form a loop with length **M.**
- Lengths of all CMS loops form a multiset **L = {M, …}.**

Examples for order N=10:

- L = {1:100} — trivial;
- L = {1:10, 2:45} — canonical, all known ODLS;
- L = {4:25} — rare 1-CF loop-4 combinatorial structures;
- L = {1:10, 3:30} — ???
- L = {1:X, …}, X>N — incorrect (hasn’t ODLS).

Length of loop of ODLS =
= LCM(L1, L2, …, Lk) of CMS loops



Different properties of CMS: example 1

CMS[0] = 0
CMS[1] = 1
CMS[2] = 2
CMS[3] = 3
CMS[4] = 4
CMS[5] = 5
CMS[6] = 6
CMS[7] = 7
CMS[8] = 8
CMS[9] = 9
CMS[10] = 30
CMS[11] = 95
CMS[12] = 32
CMS[13] = 35
CMS[14] = 48
CMS[15] = 92
CMS[16] = 93
CMS[17] = 78
CMS[18] = 65
CMS[19] = 45
CMS[20] = 31
CMS[21] = 51
CMS[22] = 83
CMS[23] = 80
CMS[24] = 47
CMS[25] = 39
CMS[26] = 72
CMS[27] = 89
CMS[28] = 40
CMS[29] = 82
CMS[30] = 10
CMS[31] = 20
CMS[32] = 12
CMS[33] = 34
CMS[34] = 33
CMS[35] = 13
CMS[36] = 59
CMS[37] = 60
CMS[38] = 50
CMS[39] = 25
CMS[40] = 28
CMS[41] = 90
CMS[42] = 53
CMS[43] = 67
CMS[44] = 94
CMS[45] = 19
CMS[46] = 64
CMS[47] = 24
CMS[48] = 14
CMS[49] = 62
CMS[50] = 38
CMS[51] = 21
CMS[52] = 70
CMS[53] = 42
CMS[54] = 88
CMS[55] = 99
CMS[56] = 76
CMS[57] = 77
CMS[58] = 74
CMS[59] = 36
CMS[60] = 37
CMS[61] = 91
CMS[62] = 49
CMS[63] = 73
CMS[64] = 46
CMS[65] = 18
CMS[66] = 84
CMS[67] = 43
CMS[68] = 69
CMS[69] = 68
CMS[70] = 52
CMS[71] = 81
CMS[72] = 26
CMS[73] = 63
CMS[74] = 58
CMS[75] = 79
CMS[76] = 56
CMS[77] = 57
CMS[78] = 17
CMS[79] = 75
CMS[80] = 23
CMS[81] = 71
CMS[82] = 29
CMS[83] = 22
CMS[84] = 66
CMS[85] = 96
CMS[86] = 87
CMS[87] = 86
CMS[88] = 54
CMS[89] = 27
CMS[90] = 41
CMS[91] = 61
CMS[92] = 15
CMS[93] = 16
CMS[94] = 44
CMS[95] = 11
CMS[96] = 85
CMS[97] = 98
CMS[98] = 97
CMS[99] = 55

0	1	2	3	4	5	6	7	8	9
1	2	0	4	3	6	5	9	7	8
2	0	1	5	6	3	4	8	9	7
3	4	5	7	9	8	1	6	2	0
5	7	4	9	8	0	2	1	6	3
6	9	8	1	2	4	7	3	0	5
8	3	7	6	0	1	9	2	5	4
9	6	3	0	7	2	8	5	4	1
7	5	6	8	1	9	0	4	3	2
4	8	9	2	5	7	3	0	1	6

1	3	4	5	6	7	0	8	9	2
5	8	7	9	0	2	4	6	3	1
6	2	9	8	3	1	5	4	7	0
3	4	1	2	8	6	7	9	0	5
2	6	3	4	7	9	1	0	5	8
4	1	2	6	5	0	9	7	8	3
0	9	5	1	4	8	3	2	6	7
9	7	6	0	1	3	8	5	2	4
7	0	8	3	2	5	6	1	4	9
8	5	0	7	9	4	2	3	1	6

Searching for ODLS corresponding for CMS:
82 hours per ODLS pair
(**~10 times slower than Euler-Parker**)

Different properties of CMS: example 2

CMS[0] = 0
CMS[1] = 44
CMS[2] = 77
CMS[3] = 38
CMS[4] = 97
CMS[5] = 47
CMS[6] = 16
CMS[7] = 43
CMS[8] = 84
CMS[9] = 9
CMS[10] = 10
CMS[11] = 23
CMS[12] = 36
CMS[13] = 26
CMS[14] = 31
CMS[15] = 83
CMS[16] = 6
CMS[17] = 98
CMS[18] = 79
CMS[19] = 80
CMS[20] = 41
CMS[21] = 21
CMS[22] = 39
CMS[23] = 11
CMS[24] = 51
CMS[25] = 74
CMS[26] = 13
CMS[27] = 75
CMS[28] = 49
CMS[29] = 95
CMS[30] = 94
CMS[31] = 14
CMS[32] = 76
CMS[33] = 33
CMS[34] = 42
CMS[35] = 45
CMS[36] = 12
CMS[37] = 91
CMS[38] = 3
CMS[39] = 22
CMS[40] = 60
CMS[41] = 20
CMS[42] = 34
CMS[43] = 7
CMS[44] = 1
CMS[45] = 35
CMS[46] = 58
CMS[47] = 5
CMS[48] = 52
CMS[49] = 28
CMS[50] = 50
CMS[51] = 24
CMS[52] = 48
CMS[53] = 54
CMS[54] = 53
CMS[55] = 87
CMS[56] = 93
CMS[57] = 64
CMS[58] = 46
CMS[59] = 73
CMS[60] = 40
CMS[61] = 67
CMS[62] = 70
CMS[63] = 86
CMS[64] = 57
CMS[65] = 96
CMS[66] = 99
CMS[67] = 61
CMS[68] = 78
CMS[69] = 71
CMS[70] = 62
CMS[71] = 69
CMS[72] = 72
CMS[73] = 59
CMS[74] = 25
CMS[75] = 27
CMS[76] = 32
CMS[77] = 2
CMS[78] = 68
CMS[79] = 18
CMS[80] = 19
CMS[81] = 85
CMS[82] = 82
CMS[83] = 15
CMS[84] = 8
CMS[85] = 81
CMS[86] = 63
CMS[87] = 55
CMS[88] = 92
CMS[89] = 89
CMS[90] = 90
CMS[91] = 37
CMS[92] = 88
CMS[93] = 56
CMS[94] = 30
CMS[95] = 29
CMS[96] = 65
CMS[97] = 4
CMS[98] = 17
CMS[99] = 66

0	6	5	7	4	8	9	3	2	1
2	1	8	6	5	9	4	0	7	3
4	3	9	8	7	1	5	2	0	6
8	7	3	4	9	5	0	1	6	2
1	8	0	9	2	3	6	7	4	5
5	9	2	3	8	6	7	4	1	0
7	0	4	5	1	2	3	6	8	9
3	5	6	2	0	7	1	8	9	4
6	4	7	1	3	0	2	9	5	8
9	2	1	0	6	4	8	5	3	7

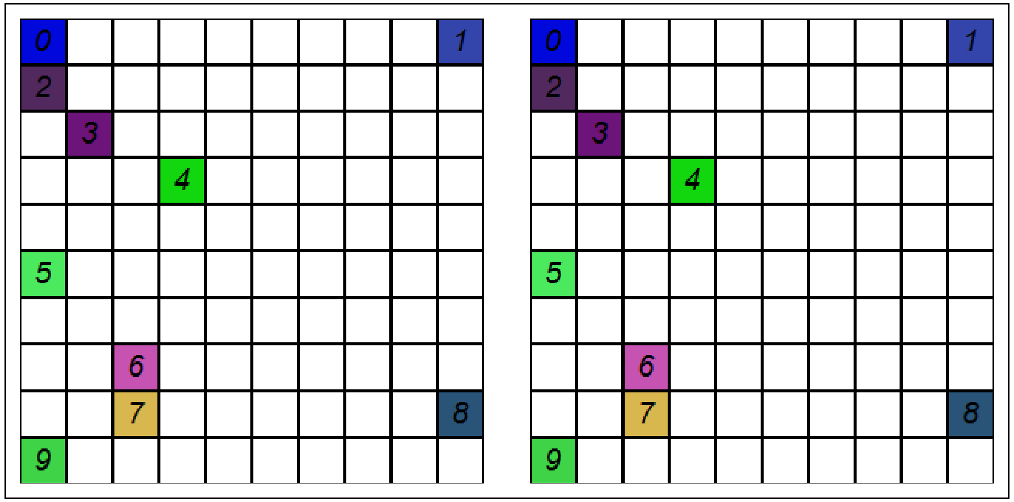
0	5	6	7	2	3	4	1	8	9
5	6	0	2	3	9	1	8	4	7
6	8	5	9	1	0	7	3	2	4
7	2	9	4	0	8	6	5	3	1
3	4	1	8	7	2	9	6	5	0
2	3	4	6	8	1	0	9	7	5
9	7	8	5	4	6	3	0	1	2
4	1	7	0	9	5	8	2	6	3
8	0	3	1	5	4	2	7	9	6
1	9	2	3	6	7	5	4	0	8

Searching for ODLS corresponding for CMS:
28 seconds per ODLS pair
(**~1000 times faster than Euler-Parker**)

- Nested loops implementation?
- GPU/Phi/FPGA implementation?
- SAT implementation?

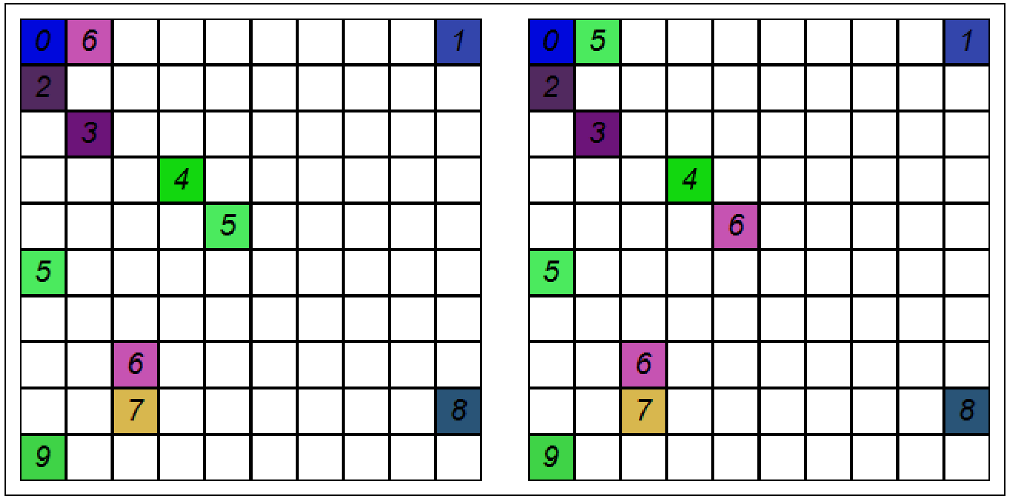
CMS filling example: fixed points

CMS[0] = 0
CMS[1] = 44
CMS[2] = 77
CMS[3] = 38
CMS[4] = 97
CMS[5] = 47
CMS[6] = 16
CMS[7] = 43
CMS[8] = 84
CMS[9] = 9
CMS[10] = 10
CMS[11] = 23
CMS[12] = 36
CMS[13] = 26
CMS[14] = 31
CMS[15] = 83
CMS[16] = 6
CMS[17] = 98
CMS[18] = 79
CMS[19] = 80
CMS[20] = 41
CMS[21] = 21
CMS[22] = 39
CMS[23] = 11
CMS[24] = 51
CMS[25] = 74
CMS[26] = 13
CMS[27] = 75
CMS[28] = 49
CMS[29] = 95
CMS[30] = 94
CMS[31] = 14
CMS[32] = 76
CMS[33] = 33
CMS[34] = 42
CMS[35] = 45
CMS[36] = 12
CMS[37] = 91
CMS[38] = 3
CMS[39] = 22
CMS[40] = 60
CMS[41] = 20
CMS[42] = 34
CMS[43] = 7
CMS[44] = 1
CMS[45] = 35
CMS[46] = 58
CMS[47] = 5
CMS[48] = 52
CMS[49] = 28
CMS[50] = 50
CMS[51] = 24
CMS[52] = 48
CMS[53] = 54
CMS[54] = 53
CMS[55] = 87
CMS[56] = 93
CMS[57] = 64
CMS[58] = 46
CMS[59] = 73
CMS[60] = 40
CMS[61] = 67
CMS[62] = 70
CMS[63] = 86
CMS[64] = 57
CMS[65] = 96
CMS[66] = 99
CMS[67] = 61
CMS[68] = 78
CMS[69] = 71
CMS[70] = 62
CMS[71] = 69
CMS[72] = 72
CMS[73] = 59
CMS[74] = 25
CMS[75] = 27
CMS[76] = 32
CMS[77] = 2
CMS[78] = 68
CMS[79] = 18
CMS[80] = 19
CMS[81] = 85
CMS[82] = 82
CMS[83] = 15
CMS[84] = 8
CMS[85] = 81
CMS[86] = 63
CMS[87] = 55
CMS[88] = 92
CMS[89] = 89
CMS[90] = 90
CMS[91] = 37
CMS[92] = 88
CMS[93] = 56
CMS[94] = 30
CMS[95] = 29
CMS[96] = 65
CMS[97] = 4
CMS[98] = 17
CMS[99] = 66



CMS filling example: fixed points + 1st pair

CMS[0] = 0
CMS[1] = 44
CMS[2] = 77
CMS[3] = 38
CMS[4] = 97
CMS[5] = 47
CMS[6] = 16
CMS[7] = 43
CMS[8] = 84
CMS[9] = 9
CMS[10] = 10
CMS[11] = 23
CMS[12] = 36
CMS[13] = 26
CMS[14] = 31
CMS[15] = 83
CMS[16] = 6
CMS[17] = 98
CMS[18] = 79
CMS[19] = 80
CMS[20] = 41
CMS[21] = 21
CMS[22] = 39
CMS[23] = 11
CMS[24] = 51
CMS[25] = 74
CMS[26] = 13
CMS[27] = 75
CMS[28] = 49
CMS[29] = 95
CMS[30] = 94
CMS[31] = 14
CMS[32] = 76
CMS[33] = 33
CMS[34] = 42
CMS[35] = 45
CMS[36] = 12
CMS[37] = 91
CMS[38] = 3
CMS[39] = 22
CMS[40] = 60
CMS[41] = 20
CMS[42] = 34
CMS[43] = 7
CMS[44] = 1
CMS[45] = 35
CMS[46] = 58
CMS[47] = 5
CMS[48] = 52
CMS[49] = 28
CMS[50] = 50
CMS[51] = 24
CMS[52] = 48
CMS[53] = 54
CMS[54] = 53
CMS[55] = 87
CMS[56] = 93
CMS[57] = 64
CMS[58] = 46
CMS[59] = 73
CMS[60] = 40
CMS[61] = 67
CMS[62] = 70
CMS[63] = 86
CMS[64] = 57
CMS[65] = 96
CMS[66] = 99
CMS[67] = 61
CMS[68] = 78
CMS[69] = 71
CMS[70] = 62
CMS[71] = 69
CMS[72] = 72
CMS[73] = 59
CMS[74] = 25
CMS[75] = 27
CMS[76] = 32
CMS[77] = 2
CMS[78] = 68
CMS[79] = 18
CMS[80] = 19
CMS[81] = 85
CMS[82] = 82
CMS[83] = 15
CMS[84] = 8
CMS[85] = 81
CMS[86] = 63
CMS[87] = 55
CMS[88] = 92
CMS[89] = 89
CMS[90] = 90
CMS[91] = 37
CMS[92] = 88
CMS[93] = 56
CMS[94] = 30
CMS[95] = 29
CMS[96] = 65
CMS[97] = 4
CMS[98] = 17
CMS[99] = 66

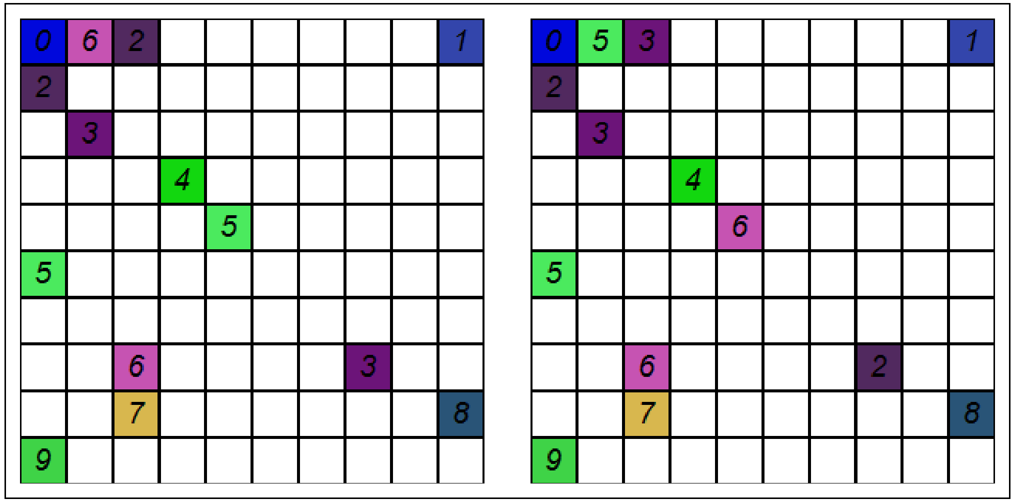


CMS filling example: fixed points + 2nd pairs

CMS[0] = 0
CMS[1] = 44
CMS[2] = 77
CMS[3] = 38
CMS[4] = 97
CMS[5] = 47
CMS[6] = 16
CMS[7] = 43
CMS[8] = 84
CMS[9] = 9
CMS[10] = 10
CMS[11] = 23
CMS[12] = 36
CMS[13] = 26
CMS[14] = 31
CMS[15] = 83
CMS[16] = 6
CMS[17] = 98
CMS[18] = 79
CMS[19] = 80
CMS[20] = 41
CMS[21] = 21
CMS[22] = 39
CMS[23] = 11
CMS[24] = 51
CMS[25] = 74
CMS[26] = 13
CMS[27] = 75
CMS[28] = 49
CMS[29] = 95
CMS[30] = 94
CMS[31] = 14
CMS[32] = 76
CMS[33] = 33

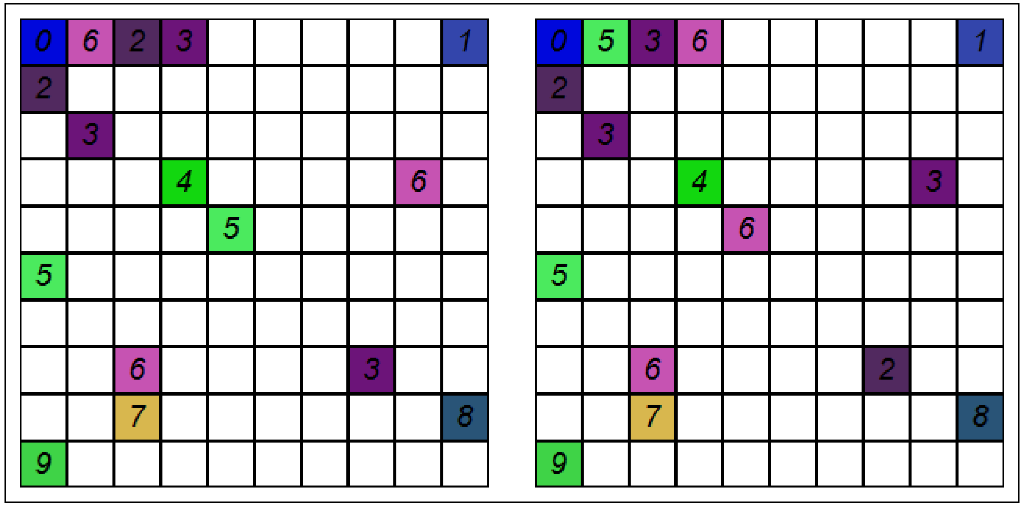
CMS[34] = 42
CMS[35] = 45
CMS[36] = 12
CMS[37] = 91
CMS[38] = 3
CMS[39] = 22
CMS[40] = 60
CMS[41] = 20
CMS[42] = 34
CMS[43] = 7
CMS[44] = 1
CMS[45] = 35
CMS[46] = 58
CMS[47] = 5
CMS[48] = 52
CMS[49] = 28
CMS[50] = 50
CMS[51] = 24
CMS[52] = 48
CMS[53] = 54
CMS[54] = 53
CMS[55] = 87
CMS[56] = 93
CMS[57] = 64
CMS[58] = 46
CMS[59] = 73
CMS[60] = 40
CMS[61] = 67
CMS[62] = 70
CMS[63] = 86
CMS[64] = 57
CMS[65] = 96
CMS[66] = 99
CMS[67] = 61

CMS[68] = 78
CMS[69] = 71
CMS[70] = 62
CMS[71] = 69
CMS[72] = 72
CMS[73] = 59
CMS[74] = 25
CMS[75] = 27
CMS[76] = 32
CMS[77] = 2
CMS[78] = 68
CMS[79] = 18
CMS[80] = 19
CMS[81] = 85
CMS[82] = 82
CMS[83] = 15
CMS[84] = 8
CMS[85] = 81
CMS[86] = 63
CMS[87] = 55
CMS[88] = 92
CMS[89] = 89
CMS[90] = 90
CMS[91] = 37
CMS[92] = 88
CMS[93] = 56
CMS[94] = 30
CMS[95] = 29
CMS[96] = 65
CMS[97] = 4
CMS[98] = 17
CMS[99] = 66



CMS filling example: fixed points + 3rd pairs

CMS[0] = 0
CMS[1] = 44
CMS[2] = 77
CMS[3] = 38
CMS[4] = 97
CMS[5] = 47
CMS[6] = 16
CMS[7] = 43
CMS[8] = 84
CMS[9] = 9
CMS[10] = 10
CMS[11] = 23
CMS[12] = 36
CMS[13] = 26
CMS[14] = 31
CMS[15] = 83
CMS[16] = 6
CMS[17] = 98
CMS[18] = 79
CMS[19] = 80
CMS[20] = 41
CMS[21] = 21
CMS[22] = 39
CMS[23] = 11
CMS[24] = 51
CMS[25] = 74
CMS[26] = 13
CMS[27] = 75
CMS[28] = 49
CMS[29] = 95
CMS[30] = 94
CMS[31] = 14
CMS[32] = 76
CMS[33] = 33
CMS[34] = 42
CMS[35] = 45
CMS[36] = 12
CMS[37] = 91
CMS[38] = 3
CMS[39] = 22
CMS[40] = 60
CMS[41] = 20
CMS[42] = 34
CMS[43] = 7
CMS[44] = 1
CMS[45] = 35
CMS[46] = 58
CMS[47] = 5
CMS[48] = 52
CMS[49] = 28
CMS[50] = 50
CMS[51] = 24
CMS[52] = 48
CMS[53] = 54
CMS[54] = 53
CMS[55] = 87
CMS[56] = 93
CMS[57] = 64
CMS[58] = 46
CMS[59] = 73
CMS[60] = 40
CMS[61] = 67
CMS[62] = 70
CMS[63] = 86
CMS[64] = 57
CMS[65] = 96
CMS[66] = 99
CMS[67] = 61
CMS[68] = 78
CMS[69] = 71
CMS[70] = 62
CMS[71] = 69
CMS[72] = 72
CMS[73] = 59
CMS[74] = 25
CMS[75] = 27
CMS[76] = 32
CMS[77] = 2
CMS[78] = 68
CMS[79] = 18
CMS[80] = 19
CMS[81] = 85
CMS[82] = 82
CMS[83] = 15
CMS[84] = 8
CMS[85] = 81
CMS[86] = 63
CMS[87] = 55
CMS[88] = 92
CMS[89] = 89
CMS[90] = 90
CMS[91] = 37
CMS[92] = 88
CMS[93] = 56
CMS[94] = 30
CMS[95] = 29
CMS[96] = 65
CMS[97] = 4
CMS[98] = 17
CMS[99] = 66



Computing experiment (Brute Force based DFS): results

- searches for random canonical CMS — **unsuccessful**;
- searches for canonical CMS from known ODLS pairs — **unsuccessful**;
- searches for ESODLS CMS — **partially successful**…

- performed within the Gerasim@Home volunteer distributed computing project on BOINC platform (https://gerasim.boinc.ru)

Computing experiment for ESODLS CMS (Brute Force based DFS): results

ESODLS: very rare objects!

0	1	2	3	4	5	6	7	8	9
1	2	0	6	7	9	8	3	4	5
3	6	7	9	8	4	2	5	1	0
4	0	8	5	2	3	7	1	9	6
5	9	4	8	3	6	0	2	7	1
7	8	6	4	0	1	3	9	5	2
6	4	5	2	1	7	9	0	3	8
9	5	1	7	6	0	4	8	2	3
2	3	9	0	5	8	1	4	6	7
8	7	3	1	9	2	5	6	0	4

06.08.2019

1CF Loop-4 (re-find in different WU's)

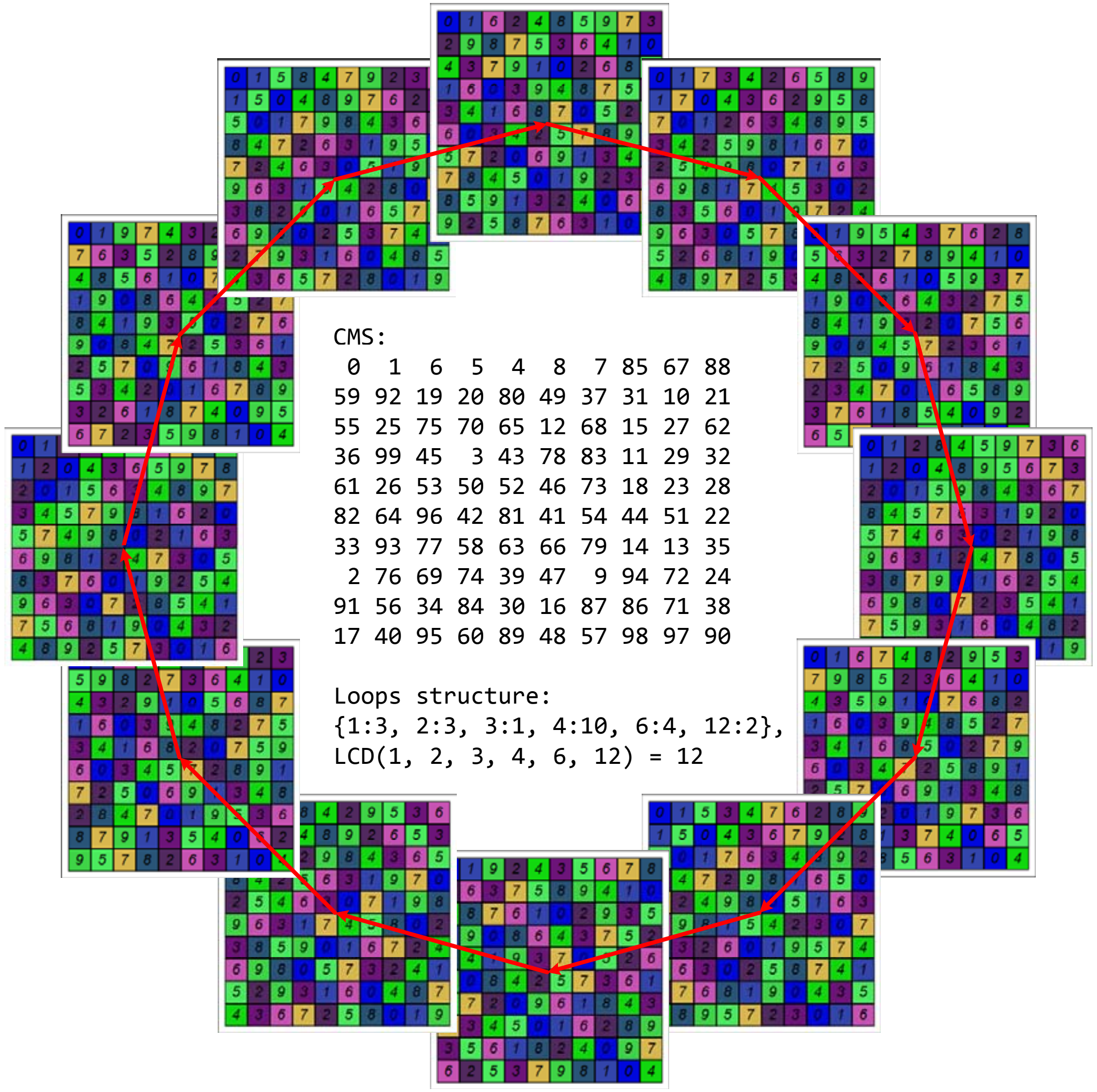
0	1	2	3	4	5	6	7	8	9
1	2	0	4	5	7	9	8	6	3
5	0	1	6	3	9	8	2	4	7
9	3	5	8	2	1	7	4	0	6
4	6	3	5	7	8	0	9	2	1
8	4	6	9	1	3	2	5	7	0
7	8	9	0	6	4	5	1	3	2
2	9	4	7	8	0	3	6	1	5
6	5	7	1	0	2	4	3	9	8
3	7	8	2	9	6	1	0	5	4

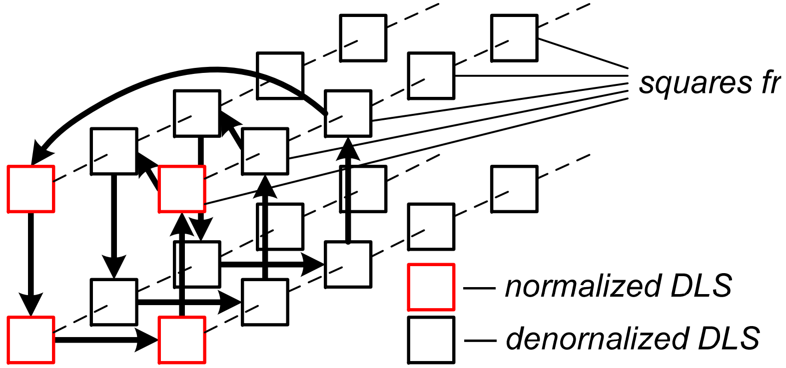
09.08.2019

1CF Once (**new!**)

- +6 additional 1 CF Onces (**new!**) and 15 re-find 1 CF Onces;
- totally 33240 ESODLS CFs (a(10)≥33240 in https://oeis.org/A309210).

Getting CMS from known ODLS pairs





Input: ODLS pair (A, B).
CMS = get_cms(A, B);
new_pairs =
 cms_based_search(CMS);

- +6 additional 1 CF Onces (**new!**) and 15 re-find 1 CF Onces;
- totally 33240 ESODLS CFs (a(10)≥33240 in https://oeis.org/A309210).

SAT approach

Boolean satisfiability problem (*SAT*) - for an arbitrary Boolean formula to determine if there exists such assignment of Boolean variables from this formula that makes it true.

Usually, a formula in considered in the Conjunctive Normal Form (CNF).

An example of CNF with 3 clauses over 5 variables:

$$C = (x_1 \vee \overline{x_2}) \cdot (x_2 \vee x_3 \vee \overline{x_4}) \cdot (\overline{x_3} \vee x_4 \vee \overline{x_5})$$

This CNF is satisfiable, e.g., on (11001).

SAT encoding

The following ESODLS CMS of order 10 were studied via SAT: CMS_{3407}, CMS_{4951}, and CMS_{1234}.

For each of them, a CNF with 436 450 clauses over 2 000 variables was constructed.

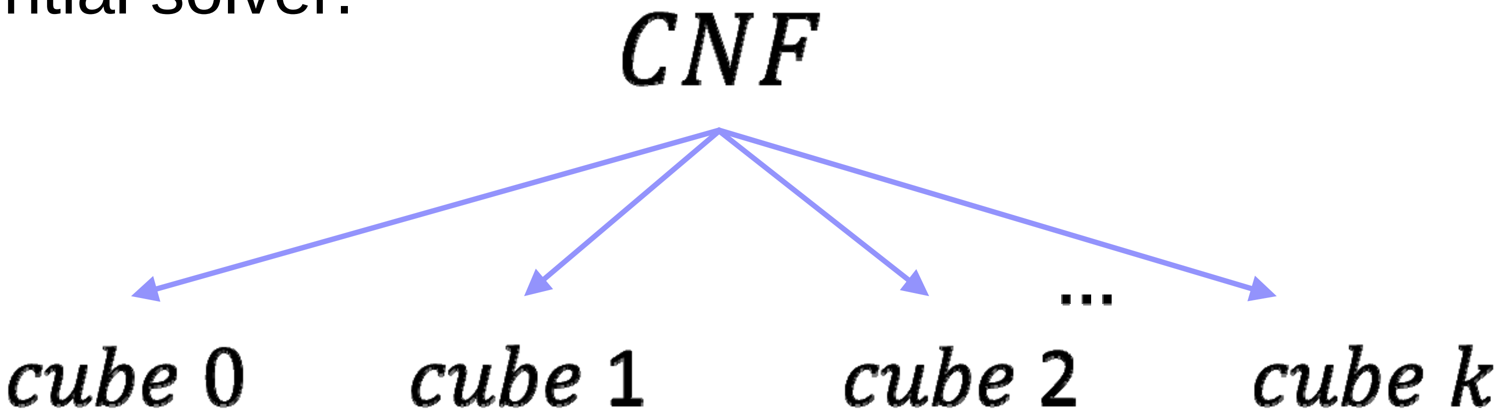
If a satisfying assignment is found for such a CNF, then for the corresponding CMS a pair of MODLS can be easily constructed based on this assignment.

Each CNF has 2 000 variables and 436 450 clauses. It turned out, that they are too hard for sequential SAT solvers.

Hard SAT instances are usually solved in parallel.

One of the possible approaches: Cube-and-Conquer.

For a given CNF, *cubes* are constructed, then on the base of each cube a simplified CNF is made and solved via a sequential solver.



Cube-and-Conquer and BOINC

- 2 265 747, 2 660 949, and 2 359 952 cubes were generated for CMS_{3407}, CMS_{4951}, and CMS_{1234} respectively.
- BOINC project RakeSearch.
- 364 334 workunits.
- Time limit of 3 minutes
- Started 20/07/2020 and ended 09/09/2020.
- 1 611 computers of 625 volunteers took part in it.
- On 29 cubes, satisfying assignments were found (interrupted on the remaining ones due to the time limit).
- 18 pairs of MODLS of order 10 were found for CMS_{3407} and 11 ones for CMS_{4951}.

Related work

Collecting CFs and new combinatorial structures search:

- triple of MODLS (is it exist?)
- different combinatorial structures for order 10?
- deep search for ESODLS using CMS (triple of ESODLS MODLS exist or not?).

Effective GPU implementation of transversal, cover and CMS based algorithms?

Enumeration problems (OEIS):

- expanding current sequences
- enumerating DLS and ODLS of special kind (string-inverse, symmetric, …) and its CFs

Pseudo triples:

- 3 kinds of pseudo triples, only 1 was investigated in details

Numerical values for DLS (OEIS):

- precise values, lower and upper constraints, spectra

Thank you for your attention!

Thanks to all the volunteers who took part in the Gerasim@home and RakeSearch projects!

WWW: http://evatutin.narod.ru, http://gerasim.boinc.ru
E-mail: evatutin@rambler.ru
LJ: http://evatutin.livejournal.com
Skype: evatutin