

Enumerating the Transversals for Diagonal Latin Squares of Small Order*

Eduard Vatutin

Southwest State University

Kursk, Russia

evatutin@rambler.ru

Stepan Kochemazov

Matrosov Institute for System Dynamics and Control Theory SB RAS

Irkutsk, Russia

veinamond@gmail.com

Oleg Zaikin

Matrosov Institute for System Dynamics and Control Theory SB RAS

Irkutsk, Russia

zaikin.icc@gmail.com

Sergey Valyaev

Internet portal BOINC.ru

Moscow, Russia

serval@boinc.ru

Abstract

In this study, we suggest an algorithm for computing minimal and maximal numbers of transversals of diagonal Latin squares. According to this algorithm, we generate all diagonal Latin squares of a particular order, and for each square construct all its transversals and diagonal transversals. With the help of this algorithm, we computed minimal and maximal numbers of transversals for diagonal Latin squares of order at most 7 on a personal computer. The experiment for diagonal Latin squares of order 8 was performed in a volunteer computing project. We also give the estimation of the runtime required to solve the corresponding problem for diagonal Latin squares of order 9.

1 Introduction

Latin squares represent a well studied combinatorial design [CD06]. They are used in various practical areas, such as cryptography, error correcting codes, etc. Also Latin squares are often used as a basis for mathematical puzzles, such as Sudoku and Magic squares. There are a lot of open problems regarding Latin squares. One class of such problems consists of enumerating problems – to enumerate all combinatorial objects (which deals

This research is partially supported by Russian Foundation for Basic Research (grants 16-07-00155-a and 17-07-00317-a) and by the Council for Grants of the President of the Russian Federation (grants No. NSH-8081.2016.9, No. MK-9445.2016.8, stipends No. SP-1829.2016.5 and No. SP-1184.2015.5).

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: E. Ivashko, A. Rumyantsev (eds.): Proceedings of the Third International Conference BOINC:FAST 2017, Petrozavodsk, Russia, August 28 - September 01, 2017, published at <http://ceur-ws.org>

with Latin squares) of the particular kind. This paper is focused on one problem from the mentioned class. In particular, we develop an algorithm for enumerating transversals of diagonal Latin squares of small order.

Let us give a brief outline of the paper. In Section 2 we describe some preliminaries on diagonal Latin squares and their transversals. In Section 3 we propose an algorithm for enumerating the number of transversals of diagonal Latin squares. In Section 4 we describe the results on enumeration of the number of transversals of diagonal Latin squares of order at most 8. In the rest of the paper we discuss related work and draw conclusions.

2 Preliminaries

A Latin square (LS) of order N is a square table $A = ||a_{ij}||$, $i, j = 1, \dots, N$, filled with elements a_{ij} from some set U , $|U| = N$ (without the loss of generality, let us below assume that $U = \{0, 1, 2, \dots, N-1\}$). By definition, in each row and each column of a Latin square every element from U appears exactly once:

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

For Diagonal Latin squares (DLS) which form a special case of LS, the definition is extended by two more conditions on the elements on Latin square main diagonal and main antidiagonal:

$$\forall i, j = \overline{1, N}, i \neq j : (a_{ii} \neq a_{jj}) \wedge (a_{i, N-i+1} \neq a_{j, N-j+1}). \quad (2)$$

A diagonal Latin square is called normalized if the elements of its first row are in ascending order. It is easy to show that any DLS can be normalized by means of a bijective mapping (transposition) of elements from U . It follows from this fact that the corresponding set of diagonal Latin squares forms an equivalence class containing $N!$ members). For a number of problems (enumeration of LS and DLS [BR75, MR95, MW05, VZZ⁺16, VKZ17], finding couples and triples of (partially) orthogonal LS and DLS [ZVZM16, ZKS16, ZZKV16]) the squares within one equivalence class do not differ, since they all possess the same properties (they have or do not have an orthogonal mate, have the same number of transversals, etc.). This fact makes it possible to significantly reduce the runtime of any related computational experiment.

A set of N entries, one selected from each row and each column of a Latin square of order N such that no two entries contain the same symbol, is called a transversal. In other words, a transversal T of a Latin square $A = ||a_{ij}||$ is a set of elements $T = \{t_1, t_2, \dots, t_N\} = \{a_{i_1, j_1}, a_{i_2, j_2}, \dots, a_{i_N, j_N}\}$, in which all indices are different

$$\forall k, l = \overline{1, N}, k \neq l : (i_k \neq i_l) \wedge (j_k \neq j_l) \quad (3)$$

and also all elements are different

$$\forall k, l = \overline{1, N}, k \neq l : a_{i_k, j_k} \neq a_{i_l, j_l}. \quad (4)$$

Vectors $[i_1, i_2, \dots, i_N]$, $[j_1, j_2, \dots, j_N]$ and $[a_{i_1, j_1}, a_{i_2, j_2}, \dots, a_{i_N, j_N}]$ form transpositions of elements from U . An example of a Latin square and the set of its transversals is shown in Figure 1. It is easy to see that both diagonals are transversals ($T_1 = \{a_{11}, a_{22}, a_{33}, a_{44}, a_{55}\}$ and $T_2 = \{a_{15}, a_{24}, a_{33}, a_{42}, a_{51}\}$, and the third transversal $T_3 = \{a_{12}, a_{21}, a_{33}, a_{45}, a_{54}\}$ is a diagonal one.

0	1	2	3	4
4	2	0	1	3
1	4	3	2	0
3	0	1	4	2
2	3	4	0	1

0				
	2			
		3		
			4	
				1

				4
			1	
		3		
	0			
2				

	1			
4				
		3		
				2
			0	

Figure 1: Example of a Latin square and its transversals

We call a transversal $T^{(d)}$ diagonal if it contains exactly one element from main diagonal of a Latin square and exactly one element from its main antidiagonal (in the case of LS of odd order these elements can overlap as in the case of example at Figure 2):

$$\begin{aligned} & \forall k, l = \overline{1, N}, \forall m, n = \overline{1, N}, m \neq k, n \neq l : \\ & \rightarrow (i_k = j_k) \wedge (i_l + j_l = N + 1) \wedge (i_m = j_m) \wedge (i_n + j_n = N + 1). \end{aligned} \quad (5)$$

A diagonal transversal is always a transversal of a general kind. An example of a Latin square and a set of its diagonal transversals is shown at Figure 2.

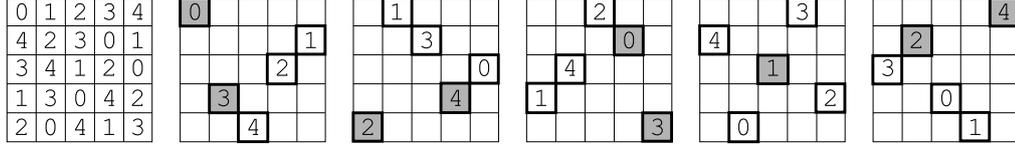
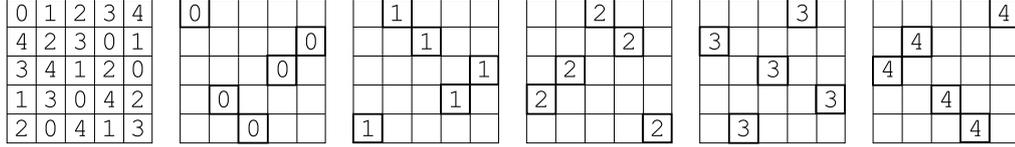


Figure 2: Example of a Latin square and its diagonal transversals. Transversals elements from main diagonal and main antidiagonal are marked with gray color



0	0	1	1	2	2	3	3	4	4
4	3	2	4	3	1	0	2	1	0
3	4	4	2	1	3	2	0	0	1
1	2	3	0	4	4	1	2	3	
2	1	0	3	4	0	1	4	3	2

Figure 3: Constructing an orthogonal mate using the set of mutually orthogonal transversals

Two transversals T_1 and T_2 are called orthogonal if they do not contain the same elements of a Latin square:

$$T_1 \perp T_2 \leftrightarrow \forall k, l, m, n = \overline{1, N} \rightarrow (a_{i_k, j_l} \in T_1) \wedge (a_{i_m, j_n} \in T_2) \wedge (a_{i_k, j_l} = a_{i_m, j_n})$$

or, in other words:

$$T_1 \perp T_2 \leftrightarrow T_1 \cap T_2 = \emptyset.$$

In the example presented at Figure 1 all transversals are not orthogonal because they contain one common element a_{33} . At the same time all diagonal transversals at Figure 2 are pairwise orthogonal.

The transversals are very useful when one needs to find systems of mutually orthogonal Latin squares or diagonal Latin squares (to which we refer below as systems of MOLS and MODLS, respectively). Two Latin squares $A = ||a_{ij}||$ and $B = ||b_{ij}||$ are called orthogonal if all ordered pairs $[a_{ij}, b_{ij}]$ are distinct. If this property is not satisfied for some of the pairs, then the corresponding Latin squares are said to be partially orthogonal, and the number of unique pairs (that do not repeat) is referred to as orthogonality characteristics. One of the most widely known currently unsolved problems in this area is to answer the question if there exists a triple of mutually orthogonal Latin squares of order 10. It is also unknown if there exists a triple of mutually orthogonal diagonal Latin squares of order 10. The best known result regarding the latter problem is the triple of squares A, B, C , where pairs A, B and A, C are orthogonal, and a pair B, C is partially orthogonal with characteristics of 74 [ZZKV16].

It is possible to construct pairs and triples of MODLS using brute force search combined with branch and bound method [LD60], or using SAT approach [ZKS16]. However, if one decides to employ transversals, it is possible to achieve much better performance compared to aforementioned approaches. In particular, it is known, that a Latin square A of order N has an orthogonal mate B if and only if A has N mutually orthogonal transversals (for diagonal Latin squares – N mutually orthogonal diagonal transversals). Having the corresponding set of N transversals it is easy to construct a square B : the elements corresponding to transversal T_i in square A are assigned with value $i - 1$ in square B (here we enumerate transversals starting from 1 and the values of Latin square elements starting from 0). At Figure 3 we show the steps leading to construction of an orthogonal mate for the diagonal Latin square from Figure 2.

3 Algorithm for Enumerating Transversals for Diagonal Latin Squares

From the constraint on the uniqueness of pairs of indices within a transversal (3) it is easy to construct the upper bound for their number, that is equal to the number $N!$ of transpositions of elements from U . The process of constructing transversals is quite similar to that of solving the widely known rooks puzzle. When we add

the constraint (4), it significantly reduces the size of the search space, thus making it possible for a particular Latin square to effectively find the set of all its transversals and then search for sets of N mutually orthogonal transversals among them.

The number of transversals significantly differs from square to square. To the best of our knowledge, there are no known analytical results regarding how the minimal and maximal number of transversals depend on the order of a Latin square. For Latin squares and non-diagonal transversals the corresponding sequence of minimal/maximal numbers of transversals depending on the order N is presented in Online Encyclopedia of Integer Sequences entries A091323 (Minimal number of transversals in a Latin square of order $2n + 1$) and A090741 (Maximum number of transversals in a Latin square of order n) [MMW06]. The question whether the maximal number of transversals for a Latin square of order 10 is 5504 is still considered an open problem [BCM⁺92].

Similar estimations of the numbers of transversals and diagonal transversals for diagonal Latin squares are unknown and can be determined in the course of computational experiment. For this purpose one needs to generate all possible DLS of order N , and for each DLS to construct the sets of its transversals and diagonal transversals. In order to generate all possible DLS it is possible to employ the highly efficient algorithm, developed by authors for this specific purpose. The algorithm has several special optimizations that take into account algorithmic features of this problem: it employs a specific order of filling Latin square elements based on the number of available variants [GB65]; it uses static data structures that help to avoid placing critical data into dynamic memory; it tracks the sets of possible variants of cell values for unfilled cells and terminates the exploration of branches of the search tree if there are square cells with empty sets of variants; it also employs auxiliary data structures (one-dimensional arrays) and bit arithmetic to determine the sets of possible variants fast [ZVZM16].

The construction of a set of diagonal transversals for a specific DLS is done using the following recurrent algorithm:

1. **Initialization.** Specify initial values for a set of transversals $S := \emptyset$, current recursion depth $d := 0$, set of available columns $C := U$, set of available elements values $E := U$.
2. **Condition for recursion end.** If $D = N$ then add current transversal T to a set of found transversals $S := S \cup \{T\}$, decrease current recursion level $d := d - 1$, go to 3c.
3. For all values $i = \overline{1, N}$ such that $(i \in C) \wedge (a_{di} \in E)$:
 - (a) Add a_{di} to transversal T : $T[d] := i$; mark i -th column as used: $C := C \setminus \{i\}$; mark a_{di} as used: $E := E \setminus \{a_{di}\}$.
 - (b) **Recurrent descent** Increase current recursion depth value: $d := d + 1$; go to 2.
 - (c) Mark i -th column as available: $C := C \cup \{i\}$; mark a_{di} as available: $E := E \cup \{a_{di}\}$.
4. **End of algorithm.**

In this algorithm we find all possible transversals of a given Latin square A . If it is necessary to find only diagonal transversals, then the corresponding checks (5) are added to the 2-nd point of the algorithm. In accordance with the branch and bound strategy, it is also possible (but not necessary) to add to paragraph 3 an additional condition (5), which checks whether the current transversal contains Latin square elements from its main diagonal and main antidiagonal. The latter can increase the algorithm by early rejection of non-diagonal transversals.

In Figure 3 we show how this algorithm works on an example of a Latin square of order 5.

4 Computational Experiment

We implemented the algorithm proposed in Section 3 and organized a computational experiment aimed at determining minimal and maximal number of transversals and diagonal transversals of diagonal Latin squares of small order.

The experiment for $1 \leq N \leq 7$ was performed on one core of Intel Core i7-4770 (Haswell). The algorithm for estimating the number of transversals for DLS of order $N = 8$ has an average performance of about 350 DLS per second (for single-threaded implementation on Delphi programming language on one core of the mentioned CPU). Taking into account the fact that the number of DLS of order 8 is 7 447 587 840 [VZZ⁺16], this experiment

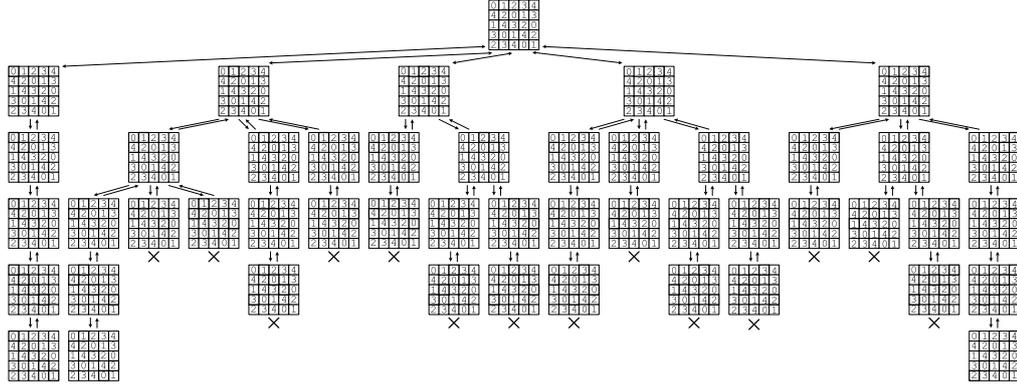


Figure 4: The example of working process of the algorithm for finding transversals

is estimated to take about 246 days on one core or about 1 month on 8 cores. We performed this experiment in the BOINC-based volunteer computing project Gerasim@home [VVT15]. In total, 3003 workunits were generated for this purpose. In each workunit the first 5 elements of the second row were fixed. Thus, an original problem was decomposed on the server by varying these 5 elements. Average runtime of each workunit was about 1 hour on 1 CPU core. The computing application had to generate all normalized DLS with fixed 5 elements (obtained from the given workunit), and determine for them the required characteristics. The results of experiments for orders $1 \leq N \leq 8$ are shown in Tables 1 and 2.

The estimation of the number of transversals for DLS of order $N = 9$ with an average performance of approximately 370 DLS per second will take about 900 years in a parallel computing system with performance of about 5 TFLOPs.

It is easy to see that for $1 \leq N \leq 8$ the maximal number of transversals for DLS coincides with that for LS (sequence A090741 in OEIS), excluding the orders 2 and 3 for which there are no DLS. It is likely that this feature will be repeated for DLS of higher orders. The minimal number of transversals is a new sequence $(1, 0, 0, 8, 3, 32, 7, 8)$ which has not yet been represented in OEIS.

When searching for diagonal transversals for DLS of order $N = 8$ the performance is about 440 DLS per second. It is significantly higher than for nondiagonal transversals. Apparently the reason for this is that there are more constraints and, because of this, less branches in the corresponding search trees. With this performance the corresponding experiment will take about 16 days using 8 threads. For $N = 9$ the corresponding performance is about 400 DLS per second and the runtime estimation is about 830 years using the same parallel computing system with performance of 5 TFLOPs.

The obtained sequences of minimal $(1, 0, 0, 4, 1, 2, 0, 0)$ and maximal number of diagonal transversals $(1, 0, 0, 4, 5, 6, 27, 120)$ have not been represented in OEIS before.

5 Related Work

Papers [BR75, MR95, MW05] report on the number of Latin squares of orders 9, 10 and 11. The authors of the present paper developed an algorithm for generating diagonal Latin squares of special kind [ZZKV16]. In [MMW06] transversals for Latin squares of order at most 9 were enumerated. Their algorithm takes into account the fact, that the space of Latin squares can be divided into isotopy classes (115 618 721 533 classes for order 9). Transversals were enumerated for each representative, this allowed to calculate the number of transversals for each isotopy class. In the present paper we don't deal with isotopy classes, because we work with diagonal Latin squares. So we had to generate all possible species of diagonal Latin squares of the considered orders.

There are several examples of application of high-performance computing to the search for combinatorial designs based on Latin squares. With the help of a computing cluster there was proven that there is no finite projective plane of order 10 [LTS89]. In the volunteer computing project SAT@home several dozen pairs of mutually orthogonal diagonal Latin squares were found [ZKS16]. In the volunteer computing project Gerasim@home, diagonal Latin squares of order 9 were enumerated [VKZ17]. A modern computing cluster was used to prove the hypothesis about the minimal number of clues in Sudoku [MTC14] (Sudoku problem is based on Latin squares). The volunteer computing project Sudoku@vtaiwan was used to confirm the solution of this problem [LW10].

Table 1: Minimal and maximal number of transversals for DLS.

N	Minimal number of transversals and corresponding DLS	Maximal number of transversals and corresponding DLS	Experiment runtime and average performance
1	1 (0)	1 (0)	< 1 s on the PC
2	-	-	< 1 s on the PC
3	-	-	< 1 s on the PC
4	8 $\begin{pmatrix} 0123 \\ 3210 \\ 1032 \\ 2301 \end{pmatrix}$	8 $\begin{pmatrix} 0123 \\ 3210 \\ 1032 \\ 2301 \end{pmatrix}$	< 1 s on the PC
5	3 $\begin{pmatrix} 01234 \\ 42013 \\ 14320 \\ 30142 \\ 23401 \end{pmatrix}$	15 $\begin{pmatrix} 01234 \\ 42301 \\ 34120 \\ 13042 \\ 20413 \end{pmatrix}$	< 1 s on the PC
6	32 $\begin{pmatrix} 012345 \\ 425031 \\ 351204 \\ 530412 \\ 243150 \\ 104523 \end{pmatrix}$	32 $\begin{pmatrix} 012345 \\ 425031 \\ 351204 \\ 530412 \\ 243150 \\ 104523 \end{pmatrix}$	< 1 s on the PC
7	7 $\begin{pmatrix} 0123456 \\ 6235014 \\ 4516230 \\ 2364501 \\ 1650342 \\ 5042163 \\ 3401625 \end{pmatrix}$	133 $\begin{pmatrix} 0123456 \\ 4260513 \\ 3516042 \\ 5634120 \\ 6452301 \\ 1305264 \\ 2041635 \end{pmatrix}$	4 min. 15 s. on the PC ≈ 600 DLS/s
8	8 $\begin{pmatrix} 01234567 \\ 12043756 \\ 47516023 \\ 35761204 \\ 54627130 \\ 76405312 \\ 23170645 \\ 60352471 \end{pmatrix}$	384 $\begin{pmatrix} 01234567 \\ 12306754 \\ 65741320 \\ 53412076 \\ 27065431 \\ 34157602 \\ 70623145 \\ 46570213 \end{pmatrix}$	3 days in Gerasim@home ≈ 350 DLS/s

Table 2: Minimal and maximal number of diagonal transversals for DLS.

N	Minimal number of diagonal transversals and corresponding DLS	Maximal number of diagonal transversals and corresponding DLS	Experiment runtime and average performance
1	1 (0)	1 (0)	< 1 s on the PC
2	-	-	< 1 s on the PC
3	-	-	< 1 s on the PC
4	4 $\begin{pmatrix} 0123 \\ 3210 \\ 1032 \\ 2301 \end{pmatrix}$	4 $\begin{pmatrix} 0123 \\ 3210 \\ 1032 \\ 2301 \end{pmatrix}$	< 1 s on the PC
5	1 $\begin{pmatrix} 01234 \\ 42013 \\ 14320 \\ 30142 \\ 23401 \end{pmatrix}$	5 $\begin{pmatrix} 01234 \\ 42301 \\ 34120 \\ 13042 \\ 20413 \end{pmatrix}$	< 1 s on the PC
6	2 $\begin{pmatrix} 012345 \\ 425031 \\ 351204 \\ 530412 \\ 243150 \\ 104523 \end{pmatrix}$	6 $\begin{pmatrix} 012345 \\ 425130 \\ 351204 \\ 530412 \\ 243051 \\ 104523 \end{pmatrix}$	< 1 s on the PC
7	0 $\begin{pmatrix} 0123456 \\ 6245103 \\ 5016234 \\ 3564012 \\ 2651340 \\ 4302561 \\ 1430625 \end{pmatrix}$	27 $\begin{pmatrix} 0123456 \\ 4260513 \\ 3516042 \\ 5634120 \\ 6452301 \\ 1305264 \\ 2041635 \end{pmatrix}$	4 min. 15 s. on the PC ≈ 600 DLS/s
8	0 $\begin{pmatrix} 01234567 \\ 12043756 \\ 70165423 \\ 46751032 \\ 37506241 \\ 54672310 \\ 63420175 \\ 25317604 \end{pmatrix}$	120 $\begin{pmatrix} 01234567 \\ 23016745 \\ 15407326 \\ 54761032 \\ 37625104 \\ 76543210 \\ 40152673 \\ 62370451 \end{pmatrix}$	3 days in Gerasim@home ≈ 350 DLS/s

6 Conclusions and Future Work

In the present paper we enumerated transversals for diagonal Latin squares of order at most 8. The experiments were partially performed in a volunteer computing project. In future we plan to organize computational experiments aimed at extending the obtained results for larger orders of diagonal Latin squares.

Acknowledgments

We thank all Gerasim@home volunteers, whose computers took part in the experiment.

References

- [BCM⁺92] J.W. Brown, F. Cherry, L. Most, E.T. Parker, and W.D. Wallis. Completion of the spectrum of orthogonal diagonal Latin squares. *Lecture notes in pure and applied mathematics*, 139:43–49, 1992.
- [BR75] Stanley E. Bammel and Jerome Rothstein. The number of 9×9 Latin squares. *Discrete Math.*, 11(1):93–95, January 1975.
- [CD06] Charles J. Colbourn and Jeffrey H. Dinitz. *Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2006.
- [GB65] Solomon W. Golomb and Leonard D. Baumert. Backtrack programming. *J. ACM*, 12(4):516–524, October 1965.
- [LD60] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *ECONOMETRICA*, 28(3):497–520, 1960.
- [LTS89] C.W.H. Lam, L. Thiel, and S. Swierz. The nonexistence of finite projective planes of order 10. *Canad. J. Math.*, 41:1117–1123, 1989.
- [LW10] Hung-Hsuan Lin and I-Chen Wu. Solving the minimum Sudoku problem. In *The 2010 International Conference on Technologies and Applications of Artificial Intelligence, TAAI '10*, pages 456–461, Washington, DC, USA, 2010. IEEE Computer Society.
- [MMW06] Brendan D. McKay, Jeanette C. McLeod, and Ian M. Wanless. The number of transversals in a Latin square. *Designs, Codes and Cryptography*, 40(3):269–284, 2006.
- [MR95] Brendan D. McKay and Eric Rogoyski. Latin squares of order 10. *Electr. J. Comb.*, 2(1):1–4, 1995.
- [MTC14] Gary McGuire, Bastian Tugemann, and Gilles Civario. There is no 16-clue Sudoku: Solving the Sudoku minimum number of clues problem via hitting set enumeration. *Experimental Mathematics*, 23(2):190–217, 2014.
- [MW05] Brendan D. McKay and Ian M. Wanless. On the number of Latin squares. *Annals of Combinatorics*, 9(3):335–344, oct 2005.
- [VKZ17] Eduard Vatutin, Stepan Kochemazov, and Oleg Zaikin. Applying volunteer and parallel computing for enumerating diagonal latin squares of order 9. In *Proc. of The Eleventh International Conference on Parallel Computational Technologies*, volume 753 of *Communications in Computer and Information Science*, pages 110–124. Springer, 2017.
- [VVT15] Eduard Vatutin, Sergey Valyaev, and Vitaly Titov. Comparison of sequential methods for getting separations of parallel logic control algorithms using volunteer computing. In *Second International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC:FAST 2015), Petrozavodsk, Russia, September 14-18, 2015*, volume 1502 of *CEUR-WS*, pages 37–51, 2015.
- [VZZ⁺16] Eduard Vatutin, Oleg Zaikin, Alexey Zhuravlev, Maxim Manzyuk, Stepan Kochemazov, and Vladimir Titov. Using grid systems for enumerating combinatorial objects on example of diagonal Latin squares. In *Selected Papers of the 7th International Conference Distributed Computing and Grid-technologies in Science and Education, Dubna, Russia, July 4-9, 2016*, volume 1787 of *CEUR-WS*, pages 486–490, 2016.

- [ZKS16] Oleg Zaikin, Stepan Kochemazov, and Alexander Semenov. SAT-based search for systems of diagonal Latin squares in volunteer computing project SAT@home. In Petar Biljanovic, Zeljko Butkovic, Karolj Skala, Tihana Galinac Grbac, Marina Cicin-Sain, Vlado Sruk, Slobodan Ribaric, Stjepan Gros, Boris Vrdoljak, Mladen Mauher, Edvard Tijan, and Dino Lukman, editors, *39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016, Opatija, Croatia, May 30 - June 3, 2016*, pages 277–281. IEEE, 2016.
- [ZVZM16] Oleg Zaikin, Eduard Vatutin, Aleksey Zhuravlev, and Maxim Manzyuk. Applying high-performance computing to searching for triples of partially orthogonal Latin squares of order 10. In *10th Annual International Scientific Conference on Parallel Computing Technologies Arkhangelsk, Russia, March 29-31, 2016*, volume 1576 of *CEUR-WS*, pages 155–166, 2016.
- [ZZKV16] Oleg Zaikin, Alexey Zhuravlev, Stepan Kochemazov, and Eduard Vatutin. On the construction of triples of diagonal Latin squares of order 10. *Electronic Notes in Discrete Mathematics*, 54:307–312, 2016.