Enumeration of Isotopy Classes of Diagonal Latin Squares of Small Order Using Volunteer Computing

Eduard Vatutin[1(✉)], Alexey Belyshev[2], Stepan Kochemazov[3], Oleg Zaikin[3], Natalia Nikitina[4]

[1] Southwest State University, Kursk, Russia
evatutin@rambler.ru
[2] BOINC.ru, Moscow, Russia
alexey-bell@yandex.ru
[3] Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia
veinamond@gmail.com, zaikin.icc@gmail.com
[4] Institute of Applied Mathematical Research KRC RAS, Petrozavodsk, Russia
nikitina@krc.karelia.ru

Abstract. The paper is devoted to discovering new features of diagonal Latin squares of small order. We present an algorithm, based on a special kind of transformations, that constructs a canonical form of a given diagonal Latin square. Each canonical form corresponds to one isotopy class of diagonal Latin squares. The algorithm was implemented and used to enumerate the isotopy classes of diagonal Latin squares of order at most 8. For order 8 the computational experiment was conducted in a volunteer computing project. The algorithm was also used to estimate how long it would take to enumerate the isotopy classes of diagonal Latin squares of order 9 in the same volunteer computing project.

Keywords: volunteer computing, combinatorics, Latin square, diagonal Latin square, enumeration

1 Introduction

There exist a number of problems in which it is required to enumerate combinatorial objects with specific features. The correspondence between the number of such objects and the problem's dimension can be viewed as an integer sequence. The Online Encyclopedia of Integer Sequences (OEIS) [1] contains about 300 thousands of such sequences. For some of them there exists a formula that allows to easily calculate any member of the sequence. In the cases where such formulas don't exist, the members of the sequences can sometimes be obtained as a result of a computational experiment. Often, such calculations require a large amount of computational resources, that is why high performance computing methods are required to conduct them.

Latin squares [2] are one of the most well studied combinatorial designs. They have applications both in science and industry. A *Latin square* of order N is a square table with $N \times N$ cells filled with elements from a finite set $\{0, 1, 2, \ldots, N-1\}$ in such a way, that all elements within each row and each column are distinct. If in addition to this both main diagonal and main antidiagonal contain every possible element from $\{0, 1, 2, \ldots, N-1\}$ then such Latin square is called *diagonal Latin square*.

Latin squares have practical applications in various areas, including experiment design, cryptography, error-correcting codes, scheduling. A number of open mathematical problems are associated with Latin squares, for example whether there exists a triple of mutually orthogonal Latin squares of order 10 [3]. Also of interest is the asymptotic behavior of combinatorial characteristics of Latin squares with the increase of dimension N.

There are several examples of applications of high-performance computing in order to search for combinatorial designs based on Latin squares. For example, the hypothesis about the minimal number of clues in Sudoku was first proven on a computing cluster [4].

The present study is aimed at enumeration of isotopy classes of diagonal Latin squares of small order. In other words, the goal is to construct the corresponding new integer sequence, not yet presented in OEIS. To achieve this goal, we develop a new combinatorial algorithm, implement it and employ a volunteer computing project to conduct the computational experiments.

2 Isotopy Classes and Canonical Forms of Diagonal Latin Squares

A number of combinatorial problems related to diagonal Latin squares can be solved significantly faster if only one diagonal Latin square from an equivalence class is considered. Each equivalence class contains diagonal Latin squares with equal features (for instance, the number of orthogonal mates). Such approach can be used to enumerate diagonal Latin squares [5, 6] or to construct and enumerate systems of orthogonal diagonal Latin squares and combinatorial designs based on them [7].

In the case of Latin squares, an isotopy class consists of all Latin squares that are equivalent: any Latin square from the class can be produced by any other Latin square from the class by permuting the rows, columns, or the names of the symbols of a Latin square. In the case of diagonal Latin squares, the isotopic classes cannot be constructed in the same manner because some permutations of rows and columns can lead to the violation of the constraint on uniqueness of elements on diagonals. Instead, the isotopy classes of diagonal Latin squares can be constructed using the so-called M-transformations, which were suggested by Yu. V. Chebrakov for magic squares [8] (by definition any diagonal Latin square is a magic square). The M-transformations can be divided into two types. An M-transformation of the first type is a permutation of a pair of columns, which are symmetric with respect to the middle of a square, along with the permutation of rows with the same indexes. An example of an M-transformation of the first type is shown in Fig. 1.

An M-transformation of the second type is a permutation of a pair of columns from the left half of the square, along with the permutation of a pair of columns from the right half of the square, which are symmetric to the columns from the first pair, and the corresponding permutation of two pairs of rows with the same indexes. An example of an M-transformation of the second type is shown in Fig. 2.

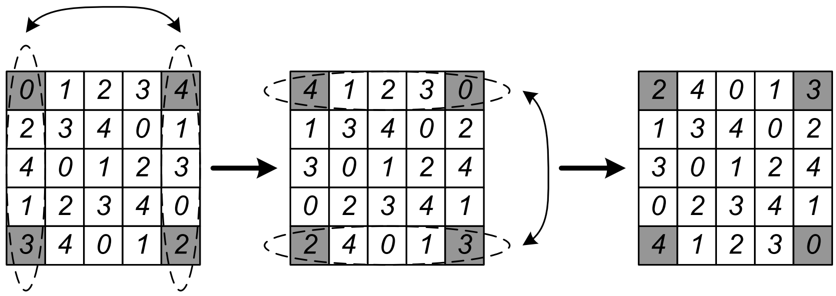

Fig. 1. An example of an M-transformation of the first type.

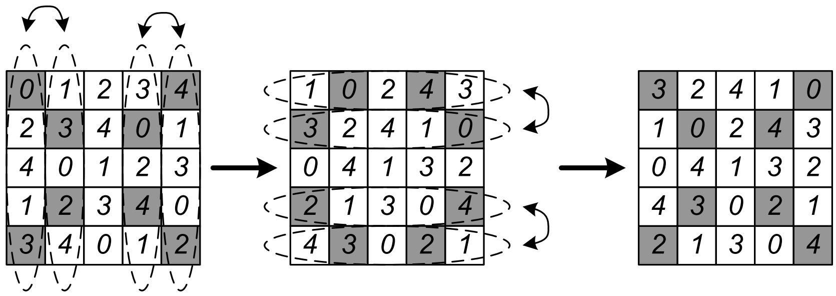

Fig. 2. An example of M-transformation of the second type.

It is easy to see, that several other equivalent transformations can be used for diagonal Latin squares: turning a square by $90 \times h$ degrees, $h \in \{1, 2, 3\}$; transposing it relative to main diagonal or main antidiagonal; mirroring it horizontally or vertically. The examples of all these transformations are shown in Fig. 3.

While all squares in an isotopy class are equal, it is convenient to choose one of them to be a representative of a class. Assume that we consider a diagonal Latin square A of order N. It can be represented as a sequence $s(A)$ with $N \times N$

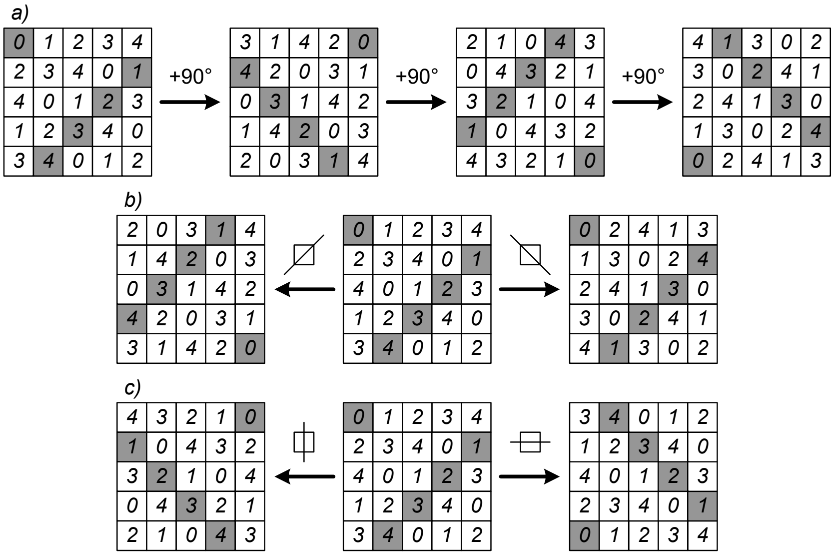

Fig. 3. Examples of equivalence transformations, which are not defined by M-transformations: turns of a square (a); transpose of a square (b); vertical and horizontal reflections (c). One of the squares transversal is marked with grey.

elements by writing all elements of A one by one from left to right, from top to bottom. Let us refer to $c(A)$ as to *string representation* of A. A *canonical form* (CF) $\tilde{A}$ of a diagonal Latin square A is a diagonal Latin square from a corresponding isotopy class $\Theta(A) = \{A_1, A_2, \ldots, A_C\}$, $A \in \Theta(A)$, $\tilde{A} \in \Theta(A)$, which has a lexicographically minimal string representation: $\forall A' \in \Theta(A) \backslash \{\tilde{A}\}$: $s(\tilde{A}) < s(A')$. If for a fixed N we construct a canonical form for every single diagonal Latin square of order N, then the number of unique canonical forms coincides with the number of isotopy classes of diagonal Latin squares of order N (in the sense outlined above), and this fact can be employed in practice.

3 Computational Experiments

The number of isotopy classes of Latin squares of order at most 11 is known [9–11] and is presented in the OEIS (sequence A040082). Meanwhile, the corresponding sequence for diagonal Latin squares has not been known before the present study. The easiest way to determine such number for a particular order N is to check each Latin square of order N whether it is a canonical form or not. To generate all diagonal Latin squares of small order, a fast generator can be used [5]. In order to construct a canonical form for a given Latin square A, the corresponding isotopy class $Q(A) = \{A_1, A_2, \ldots, A_C\}$ is built using M-transformations and other transformations preserving the contents of main diagonal and main antidiagonal of a Latin square (they were mentioned in the previous section). It should be noted, that Latin square A is a canonical form if and only if $A = argmin_{A_i \in \Theta(A)} A_i$. A strict implementation of this approach (on the Delphi language) allows to check about 1-2 diagonal Latin squares of order 10 per second, which is very slow. Note, that generator described in [5] allows to generate about 6.6 millions of diagonal Latin squares per second.

There were introduced several optimizations in the implementation of the algorithm that constructs canonical forms. They are described below, together with estimated gain. The corresponding experiments were held on a computer equipped with the Core i7-4770 CPU and DDR3 RAM. Note, that small size of the used data structures made it possible to use only CPU's L1 cache and not to use RAM.

1. Reducing the amount of usage of dynamic memory and dynamic strings: 31 diagonal Latin squares per second.
2. Employing the Johnson-Trotter algorithm [12] to form M-permutations: 56 diagonal Latin squares per second.
3. Disabling the checkups of the built-in compilator ($R, ASSERTs, etc.): 118 diagonal Latin squares per second.
4. Normalizing a diagonal Latin square in the end of the search together with its mirrored version: 277 diagonal Latin squares per second.
5. Using the horizontal reflection instead of the vertical reflection: 305 diagonal Latin squares per second.
6. Terminating the processing of a current diagonal Latin square if its string representation is less than that of an original one, high-level optimizations: 642 diagonal Latin squares per second.

It should be noted, that the aforementioned rates were obtained at one of the first sections of the search space (of all possible diagonal Latin squares, in lexicographic order), where the concentration of canonical forms is high and the corresponding checks take quite a lot of time. At the final sections of the search space the canonical forms are sparse, and the performance is several times higher because of early terminations.

Using this implementation (author: Eduard Vatutin) a computational experiment was conducted on a personal computer. As a result, the isotopy classes of diagonal Latin squares of order at most 7 were enumerated, forming the following sequence (for $1 \leq N \leq 7$, where N is an order): 1, 0, 0, 1, 2, 2, 972.

A computational experiment for order 8 was conducted in the volunteer computing project Gerasim@home [13]. The deadline of 7 days, and the quorum of 2 were used. A set of work units (WUs) was formed in such a way, that each WU contains initial values of several elements of a square. Given a particular WU, a client application on a volunteers personal computer forms all possible values of other elements of a square in order to generate all possible diagonal Latin squares (with fixed values of some elements provided by the WU). The client application enumerates all canonical forms among the generated diagonal Latin squares, and the obtained number is sent to the project server (see Fig. 4).

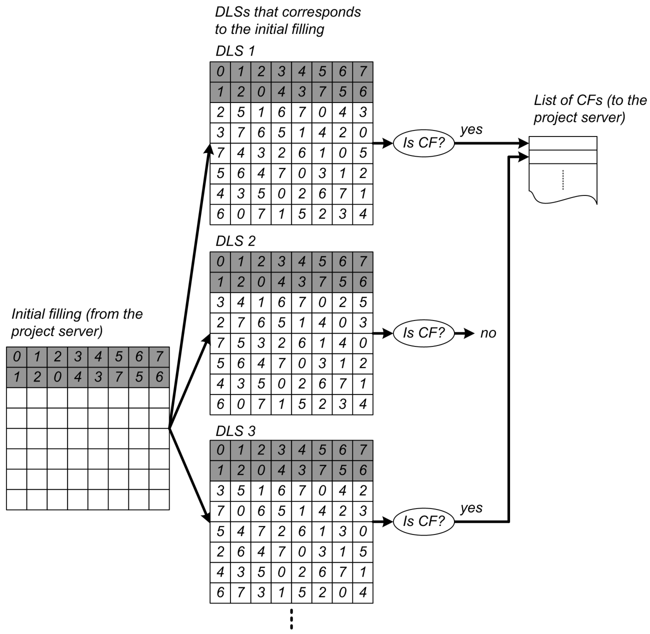

Fig. 4. Schematic representation of the calculation process on a client computer.

This experiment was conducted in 2 days. As a result, the isotopy classes of diagonal Latin squares of order 8 were enumerated, it turned out that there are 4 873 096 of them (remind, that each such class corresponds to a canonical form). This result was verified by another program implemented by Alexey Belyshev. His implementation is based on normalization of diagonal Latin squares performed based on the main diagonal. Thus, the following sequence was obtained: 1, 0, 0, 1, 2, 2, 972, 4873096. It represents the number of isotopy classes of diagonal Latin squares of order at most 8. This sequence was unknown before the present study, it has been reviewed and added to the OEIS with the number A287764.

For order 9 the number of corresponding normalized diagonal Latin squares is 505 699 465 350 758 [5]. Taking into account that an average rate of checking for canonical forms for diagonal Latin squares of order 9 is about 60 thousand squares per second, it means that to enumerate the isotopy classes for order 9 it would take about 267 years of calculations on 1 CPU core, or about 1 year in a volunteer computing project with the real performance of 1 TFLOPS/s.

In the course of the conducted experiments, the minimal and maximal sizes of isotopy classes of diagonal Latin squares of order at most 8 were also found, see Table 1. The corresponding sequences 1, 0, 0, 2, 4, 32, 32, 96 and 1, 0, 0, 2, 4, 96, 192, 1536 are new and have not been represented in the OEIS yet.

First pairs of orthogonal diagonal Latin squares of order 10 were first presented in [14]. The volunteer computing projects Gerasim@home [13] and SAT@home

Table 1. The minimal and maximal sizes of isotopy classes of diagonal Latin squares of order at most 8 and examples of corresponding squares.

N	Minimal size of isotopy class and its representative		Maximal size of isotopy class and its representative	
1	1	-	1	-
2	-	-	-	-
3	-	-	-	-
4	2	0 1 2 3 2 3 0 1 3 2 1 0 1 0 3 2	2	0 1 2 3 2 3 0 1 3 2 1 0 1 0 3 2
5	4	0 1 2 3 4 2 3 4 0 1 4 0 1 2 3 1 2 3 4 0 3 4 0 1 2	4	0 1 2 3 4 2 3 4 0 1 4 0 1 2 3 1 2 3 4 0 3 4 0 1 2
6	32	0 1 2 3 4 5 1 2 0 5 3 4 4 3 5 0 2 1 3 5 1 4 0 2 5 4 3 2 1 0 2 0 4 1 5 3	96	0 1 2 3 4 5 1 2 0 5 3 4 4 3 5 0 2 1 3 0 1 4 5 2 5 4 3 2 1 0 2 5 4 1 0 3
7	32	0 1 2 3 4 5 6 1 2 5 6 0 3 4 4 6 3 0 5 1 2 5 4 6 1 3 2 0 3 5 4 2 6 0 1 6 0 1 5 2 4 3 2 3 0 4 1 6 5	192	0 1 2 3 4 5 6 1 2 5 4 6 0 3 5 0 3 6 2 4 1 2 4 6 1 0 3 5 6 3 4 0 5 1 2 4 5 1 2 3 6 0 3 6 0 5 1 2 4
8	96	0 1 2 3 4 5 6 7 1 2 3 5 7 6 0 4 3 0 1 7 5 4 2 6 5 6 7 4 3 0 1 2 7 3 5 1 6 2 4 0 4 7 6 0 2 3 5 1 6 5 4 2 0 1 7 3 2 4 0 6 1 7 3 5	1 536	0 1 2 3 4 5 6 7 1 2 5 6 7 3 0 4 5 7 1 0 6 2 4 3 7 3 6 4 1 0 5 2 2 6 4 5 3 7 1 0 4 5 3 7 0 6 2 1 3 4 0 2 5 1 7 6 6 0 7 1 2 4 3 5

[7] have been used to find systems of mutually orthogonal diagonal Latin squares of order 10. It was done in an attempt to prove the existence or non-existence of a triple of mutually orthogonal diagonal Latin squares (MODLS) of order 10. Note, that in [15] a triple of diagonal Latin squares of order 10, that is the closest to being a triple of MODLS of order 10 found so far, was provided.

To exclude the duplication of found systems of ODLS of order 10, for each system the canonical forms of its squares are constructed, and then these canonical forms are added to a special list, in which all elements are unique. This procedure has been performed for all systems of ODLS of order 10 found so far in two mentioned volunteer computing projects. At the present moment (April 2018), this list consists of 309 thousand canonical forms of diagonal Latin squares of order 10 (see Fig. 5). It turned out, that based on these canonical forms, the triple of MODLS of order 10 cannot be constructed, and neither it is possible to construct a pseudotriple which is better than that found in [15].

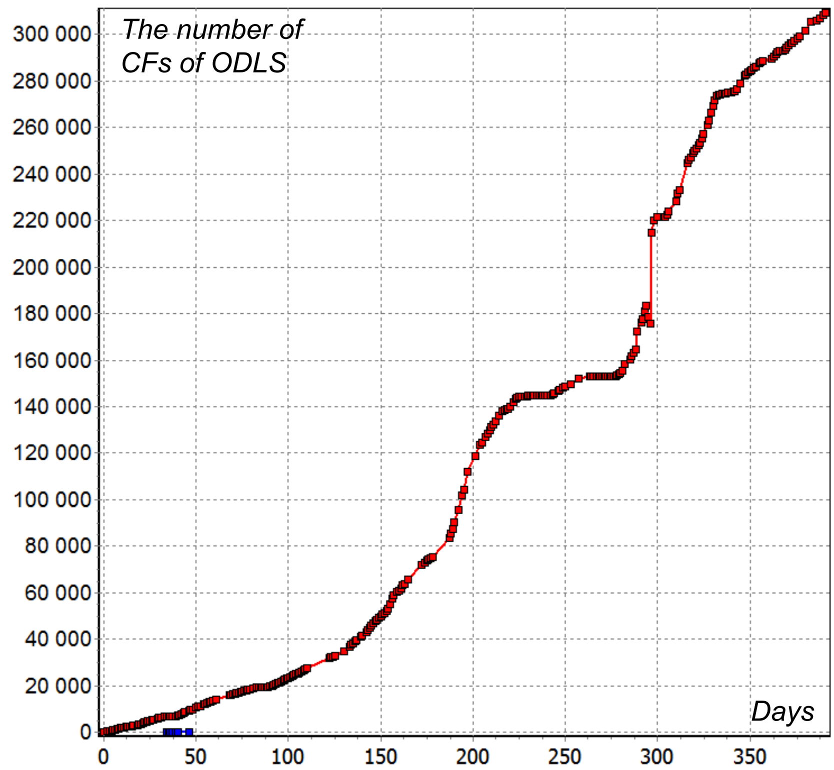

Fig. 5. The number of unique canonical forms met in systems of orthogonal diagonal Latin squares of order 10 found in Gerasim@home for the last year

4 Conclusions

Using a combinatorial algorithm, which allows to construct canonical forms, the isotopy classes of diagonal Latin squares of order at most 8 were enumerated. The experiments for order up to 7 were conducted on a computer, while for order 8 it was conducted in a volunteer computing project. It was also estimated, that isotopy classes for order 9 can be enumerated in reasonable time using the same volunteer computing project. In the nearest future we are planning to launch the corresponding experiment. It was also shown, that the canonical forms can be used to maintain a list of all systems of orthogonal diagonal Latin squares found so far.

Acknowledgements The research was partially supported by Russian Foundation for Basic Research (grants 16-07-00155-a, 17-07-00317-a, 18-07-00628-a, 18-37-00094-mol-a) and by Council for Grants of the President of the Russian Federation (stipend SP-1829.2016.5).

References

1. Sloane, N.: The on-line encyclopedia of integer sequences. https://oeis.org/
2. Colbourn, C., Dinitz, J., Wanless, I., Bennett, F., Lindner, C., Julian R. Abel, R., Finizio, N., Zhu, L., Greig, M.: Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications), chap. Latin Squares, pp. 224–265. Chapman and Hall/CRC (2006)
3. Egan, J., Wanless, I.M.: Enumeration of MOLS of small order. Math. Comput. 85(298), 799–824 (2016)
4. McGuire, G., Tugemann, B., Civario, G.: 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)
5. Vatutin, E., Kochemazov, S., Zaikin, O.: Applying volunteer and parallel computing for enumerating diagonal Latin squares of order 9. In: Proc. of The Eleventh International Conference on Parallel Computational Technologies. Communications in Computer and Information Science, vol. 753, pp. 110–124. Springer (2017)
6. Vatutin, E., Zaikin, O., Zhuravlev, A., Manzyuk, M., Kochemazov, S., Titov, V.S.: Using grid systems for enumerating combinatorial objects on example of diagonal Latin squares. In: CEUR Workshop proceedings. Selected papers of the 7th International Conference on Distributed Computing and Grid-technologies in Science and Education. vol. 1787, pp. 486–490 (2017)
7. Vatutin, E., Zaikin, O., Kochemazov, S., Valyaev, S.: Using volunteer computing to study some features of diagonal Latin squares. Open Engineering 7, 453–460 (2017)
8. Pickover, C.A.: The Zen of Magic Squares, Circles, and Stars: An Exhibition of Surprising Structures across Dimensions. Princeton University Press (2002)
9. Hulpke, A., Kaski, P., Östergård, P.: The number of Latin squares of order 11. Mathematics of Computation 80(274), 1197–1219 (2011)
10. McKay, B.D., Rogoyski, E.: Latin squares of order 10. Electr. J. Comb. 2(1), 1–4 (1995)

11. McKay, B.D., Wanless, I.M.: On the number of Latin squares. Annals of Combinatorics 9(3), 335–344 (oct 2005)
12. Trotter, H.F.: Algorithm 115: Perm. Commun. ACM 5(8), 434–435 (1962)
13. Vatutin, E., Valyaev, S., Titov, V.: 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. CEUR-WS, vol. 1502, pp. 37–51 (2015)
14. Brown, J., Cherry, F., Most, L., Parker, E., Wallis, W.: Completion of the spectrum of orthogonal diagonal Latin squares. Lecture notes in pure and applied mathematics 139, 43–49 (1992)
15. Zaikin, O., Zhuravlev, A., Kochemazov, S., Vatutin, E.: On the construction of triples of diagonal Latin squares of order 10. Electronic Notes in Discrete Mathematics 54, 307–312 (2016)