Diagonalization and Canonization of Latin Squares

Eduard Vatutin[1(✉)], Alexey Belyshev[2], Natalia Nikitina[3], Maxim Manzuk[2], Alexander Albertian[4], Ilya Kurochkin[4], Alexander Kripachev[1], and Alexey Pykhtin[1]

[1] Southwest State University, Kursk, Russia
evatutin@rambler.ru
[2] Internet Portal BOINC.Ru, Moscow, Russia
[3] Institute of Applied Mathematical Research, Karelian Research Center of the Russian Academy of Sciences, Petrozavodsk, Russia
nikitina@krc.karelia.ru
[4] Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Moscow, Russia
assa@4ip.ru

Abstract. Article aimed to the description of equivalent transformations that are allow to get at least one diagonal Latin square (DLS) from all main classes of DLS included in main class of given Latin square (LS) if they are exist. Detailed description of corresponding algorithms for LS of odd and even orders is given. Estimations for time and memory complexities are presented, algorithms are provided with detailed examples. Description of the results of diagonalization and canonization is shown. They allow to get collections of orthogonal diagonal Latin squares in a more efficient way comparing with direct usage of Euler-Paker method using volunteer distributed computing projects Gerasim@Home and RakeSearch on BOINC platform. The possibility of obtaining stronger upper and lower bounds for some numerical series in OEIS connected with DLS using suggested transformations is shown. Prospects for further application of these transformations using distributed software implementation of corresponding algorithms are outlined.

Keywords: Latin squares · Diagonal Latin squares · Orthogonal diagonal Latin squares · Diagonalization · Canonization · Isomorphism classes · OEIS · Gerasim@Home · RakeSearch · BOINC

1 Introduction

One of the widely known types of combinatorial objects is Latin squares (LS) [8,9]. An LS A of order N is a square matrix of size $N \times N$, the cells $A[x,y]$, $x, y = \overline{0, N-1}$ of which are filled with elements of some alphabet U of cardinality $|U| = N$ (for definiteness, $U = \{0, 1, \ldots, N-1\}$) in such a way that the values

V. Voevodin et al. (Eds.): RuSCDays 2023, LNCS 14389, pp. 48–61, 2023.
https://doi.org/10.1007/978-3-031-49435-2_4

are not duplicated in the rows and columns of the square (rows and columns are different permutations of the elements of the alphabet U). For diagonal Latin Squares (DLS), an additional requirement is added that prohibits duplication of values on the main and secondary diagonals of the square (DLS diagonals are transversals). A number of open mathematical problems are associated with LS and DLS, such as the enumeration of squares of general and special form [3,12,14,18,21], the search for squares with an extremal value of one of the numerical characteristics (the number of transversals, intercalates, loops, orthogonal co-squares, etc.) [4,7,13,19], the construction of spectra of numerical characteristics [5], the search for analytical formulas for the corresponding numerical series, the construction of combinatorial structures from LS/DLS on the set of a binary orthogonality relation [20], etc.

When processing low-order squares (usually, $N \leq 7$), at the current level of development of computer technology and telecommunications, it is permissible to use the exhaustive (Brute Force) enumeration method to enumerate LS/DLS of a given type in combination with a corresponding post-processor, for which the computing resources of modern multi-core processors are sufficient. As the dimension of the problem N increases, a "combinatorial explosion" is observed, and the computational complexity of the corresponding algorithms drastically increases, which forces us to use a number of features of the problem (for example, partitioning into isomorphism classes [17]) and develop highly efficient software implementations focused on execution on computing facilities with parallel architectures (computing clusters, supercomputers, grids). From the point of view of parallel programming, combinatorial tasks are weakly-coupled, which allows splitting the original task into independent subtasks (work units, abbr. WU) in accordance with the "bag-of-tasks" principle, followed by their launch on the nodes of grid systems that have received a wide distribution in recent decades due to the active development of the Internet (both in terms of availability and throughput of the corresponding communication channels). The largest example of grid systems today is the BOINC platform [2], which includes several dozen projects from various fields of science, where millions of users (crunchers) participate all over the globe, providing free computing resources of their desktop computers and mobile devices.

In this paper, we address the implementation features of the LS diagonalization and canonization transformations, which work in conjunction with a number of other LS/DLS enumeration and post-processing algorithms, which made it possible to obtain a number of new numerical estimates for the numerical characteristics of the DLS. The corresponding calculations were performed using a computation module oriented to execution under BOINC within the framework of the volunteer distributed computing projects Gerasim@Home[1] and RakeSearch[2].

[1] http://gerasim.boinc.ru.
[2] https://rake.boincfast.ru/rakesearch.

2 Basic Concepts and Definitions

In order to preserve the rigor of the presentation of the further material of the paper, it is necessary to introduce a number of concepts and definitions. An *intercalate* in a LS is a LS of order 2×2 standing at the intersection of a certain pair of rows and columns. A *transversal* T_i in a LS is a set of N cells in which all row numbers, all column numbers and all values are different. The set of transversals of a LS will be denoted as T. A *diagonal transversal* in a DLS is a transversal in which there is one element from both main and secondary diagonals (these elements can coincide in the central cell for a DLS of odd order). The *canonical form* (CF) of a DLS is [16] the lexicographically minimal string representation of the DLS within the corresponding main class of the DLS. A pair of LS/DLS A and B is called *orthogonal* (abbr. OLS/ODLS) if all ordered pairs of values $(A[x, y], B[x, y])$, $x, y = \overline{0, N-1}$ in its composition are unique.

The number of transversals, intercalates, ODLSs and other objects in a given square is one of the numerical characteristics that have a minimum and maximum value, as well as the corresponding set (*spectrum*) of possible values. Integer numerical sequences obtained for the selected numerical characteristic with increasing problem dimension N are of fundamental importance and are collected within the framework of the corresponding Online Encyclopedia of Integer Sequences (OEIS) [15].

3 Embedding of LS and DLS Isomorphism Classes

For a LS, it is permissible to use equivalent transformations of *isotopy* which include permutation of rows, columns, and renumbering of elements. These transformations make it possible to set an equivalence relation on the set of LSs and divide them into equivalence classes called *isotopic* ones. The cardinality of the isotopy classes does not exceed $(N!)^3$. Each isotopy class can contain several DLS isomorphism classes (*main classes* of DLS). The main classes of DLS, in turn, can be divided into subclasses, however, in the context of this paper, this issue is of no interest and is not considered. The purpose of the diagonalization transformation considered in the paper is to obtain at least one DLS from each of the main DLS classes for a given initial LS. In order to avoid duplication, DLS obtained during diagonalization are transferred to CF (being canonized), which are subsequently collected.

One of 6 parastrophic transformations can be applied to the squares in the LS isotopy class, as a result of which 6 parastrophic slices (isotopy classes) will be obtained, which together form the *main class* of LS (not to be confused with the main class of DLS considered above). If a LS has generalized symmetries (automorphisms), some parastrophic slices, then LS in the isotopy classes and/or main classes of the DLS may coincide.

Schematically, the hierarchy of isomorphism classes of LS and DLS is shown in Fig. 1.

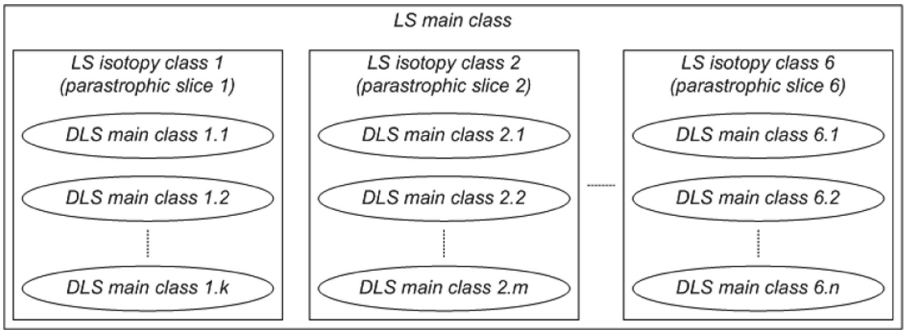

Fig. 1. Hierarchy of isomorphism classes of LS and DLS.

Remark. The depiction of the nested hierarchy of isomorphism classes shown in Fig. 1 is simplified: one main class of DLS can be nested in several paratopy classes of LS, since in the paratopy classes of LS it is not allowed to use rotations or transpositions of the square, unlike the main classes of DLS.

The isomorphism classes considered above are characterized by the presence of various invariants. For example, the invariants of the main class of LS are the number of intercalates, the number of transversals, and the number of OLS, while the invariant of the main class of DLS is the number of diagonal transversals and the number of ODLS. The lexicographically minimal CF of the DLS obtained as a result of the canonization of a given LS, if it exists, is a complete invariant of the main class of LS.

4 Diagonalization and Canonization of Latin Squares

Under the *diagonalization of a LS* we mean the procedure of targeted permutation of rows and columns aimed at obtaining at least one correct DLS from each main class of DLS in the composition of the corresponding isotopy class of LS. The algorithm for performing all possible combinations of permutations of rows and columns has an asymptotic time complexity of the order $t \simeq O((N!)^2)$ and is not applicable for practically important orders of the LS. The diagonalization procedure considered in this paper, as will be shown below, has polynomial time asymptotics both on the order of the square N and on the number of transversals $|T|$.

A schematic description of the diagonalization procedure is given below.

1. Find a pair of transversals T_i and T_j symmetrically placed by Brown [6] in the given LS A.
2. By the targeted permutation of the rows and columns in LS A, set transversal T_i to the main diagonal to obtain LS A'. Transversal T_j in LS A will be transformed into a transversal T'_j in LS A'.

3. By the targeted permutation of the rows and columns in LS A', set transversal T_j' to the secondary diagonal, leaving transversal T_i on the main diagonal, obtaining the resulting DLS A''.

An example schematically explaining the process of setting transversal T_i to the main diagonal is shown in Fig. 2.

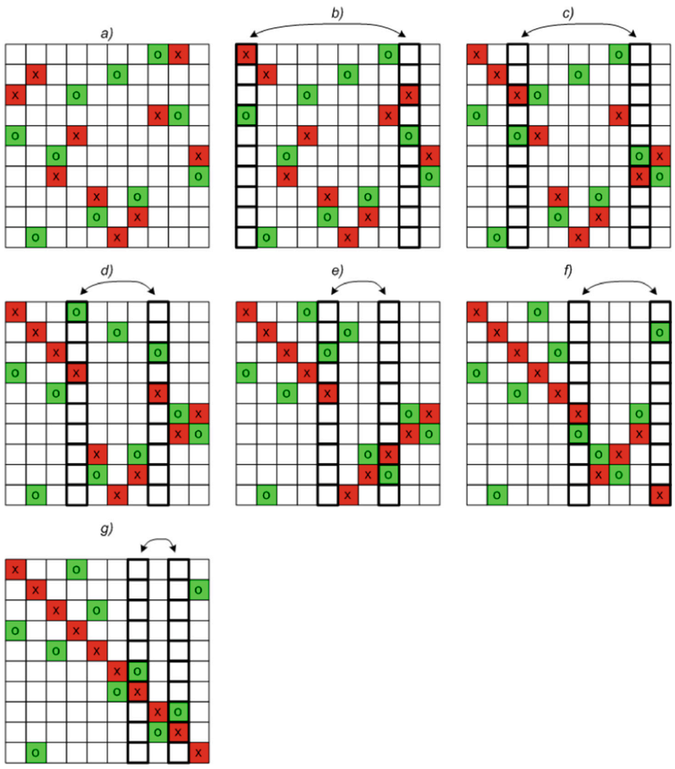

Fig. 2. Setting the transversal T_i of LS A (the elements are indicated by the symbol "x") to the main diagonal by the targeted permutation of the columns to obtain the LS A'.

After setting transversal T_i to the main diagonal, the elements of transversal T_j are located symmetrically with respect to it (see Fig. 2g). An example schematically explaining the process of setting transversal T_j to a secondary diagonal is shown in Fig. 3.

Let us consider the algorithms of the corresponding transformations in more detail. We will say that a cell of LS with coordinates $[x, y]$ belongs to the transversal $T_k \in T$ if $T_k[x] = y$ (the transversal is a one-dimensional array (permutation) in terms of programming languages, see the example in Fig. 4).

The condition that the transversals T_i and T_j are placed symmetrically by Brown can be formulated as follows: for each row x_1 of the LS containing the

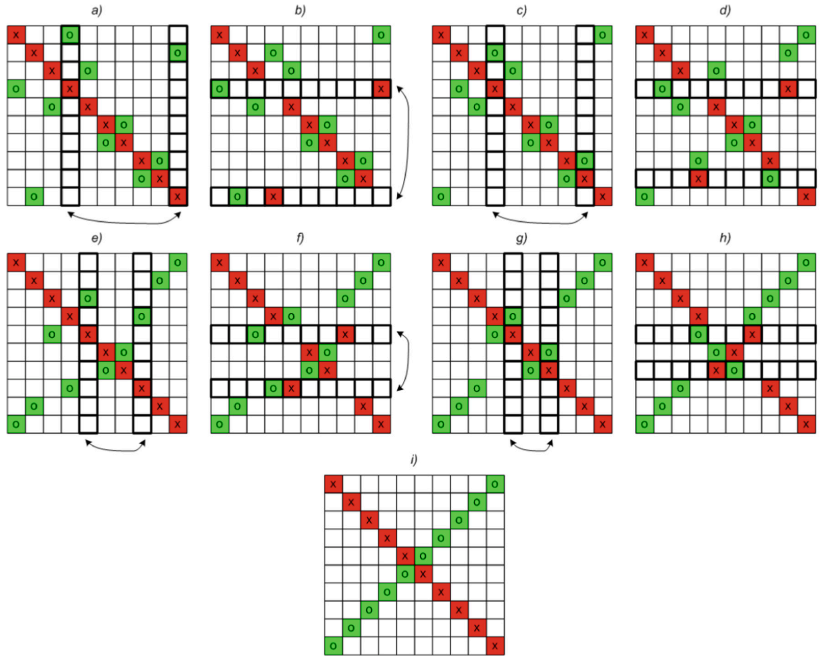

Fig. 3. Setting the transversal T'_j of LS A' (the elements are indicated by the symbol "o") to the secondary diagonal by the targeted permutation of the rows and columns to obtain the DLS A''.

elements $[x_1, T_i[x_1]] = [x_1, y_1]$ and $[x_1, T_j[x_1]] = [x_1, y_2]$ of the considered pair of transversals, there is a row x_2 such that it will contain the elements of the transversals $[x_2, T_i[x_2]] = [x_2, y_1]$ and $[x_2, T_j[x_2]] = [x_2, y_2]$. Moreover, $T_i[x_1] = T_j[x_2] = y_1$ and $T_j[x_1] = T_i[x_2] = y_2$.

In other words, the elements $[x_1, T_i[x_1]]$, $[x_1, T_j[x_1]]$, $[x_2, T_j[x_2]]$ and $[x_2, T_i[x_2]]$ form a rectangle in the LS (see Fig. 5).

The algorithm for checking a pair of transversals for symmetry by Brown is reduced to searching for a row x_2 that satisfies the above condition for a given row x_1 among all rows not yet considered and is presented below.

1. Let the set of considered rows $S := \emptyset$; the number of the first row $x_1 := 0$.
2. If the current first row has already been considered before ($x_1 \in S$), go to step 6.
3. Find row x_2 that satisfies the Brown symmetry condition considered above in a pair with row x_2.
4. If row x_2 was not found, then return the result "the pair of transversals is not symmetric by Brown" ($r := 0$); go to step 9.
5. Mark rows x_1 and x_2 as considered $S := S \bigcup \{x_1, x_2\}$.
6. Consider next row: $x_1 = x_1 + 1$.
7. If $x_1 < N$, go to step 2.

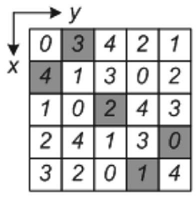

Fig. 4. An example of a DLS of order 5 and a transversal $T_k = [1, 0, 2, 4, 3]$.

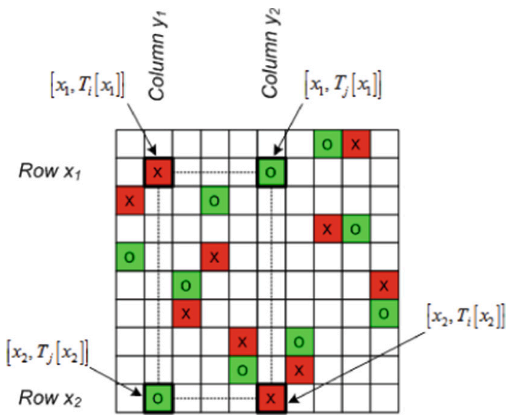

Fig. 5. An illustration explaining the condition for the symmetry of a pair of transversals by Brown (a pair of transversals, denoted by the symbols "x" and "o", is taken from the example considered above, see Figs. 2 and 3).

8. Return the result "the pair of transversals is symmetric by Brown" ($r := 1$).
9. End of the algorithm.

When implemented explicitly, the algorithm requires viewing $\frac{N}{2}$ rows x_1 of the LS, for each of which a search is made for the corresponding row x_2 (step 3), performed in linear time, which leads to time asymptotic complexity $t \simeq O(N^2)$. From the above condition $T_j[x_2] = y_1$, it follows that $x_2 = T_j^{-1}[y_1] = T_j^{-1}[T_i[x_1]]$ (another option is $x_2 = T_i^{-1}[y_2] = T_i^{-1}[T_j[x_1]]$ as a consequence of $T_j[x_2] = y_1$), where T_k^{-1} is the permutation inverse to T_k, which allows finding the row number x_2 in time independent of N, and reduces the time complexity of the algorithm to $t \simeq O(N)$, and the algorithm itself can be reduced to checking one of the conditions $T_j[T_i^{-1}[T_j[x_1]]] = y_1 = T_i[x_1]$ or $T_i[T_j^{-1}[T_i[x_1]]] = y_2 = T_j[x_1]$ for all row numbers $x_1 = \overline{0, N-1}$.

It is easy to see that when working with a LS of an even order, transversals placed symmetrically by Brown should not intersect due to the fact that the main and secondary diagonals also do not intersect, for a LS of an odd order there must be exactly one intersection point, which will subsequently be set to the center of the square.

Let us consider the algorithms for setting transversals placed symmetrically by Brown to the diagonal using the example of a LS of even order. As noted above, first it is necessary to set the transversal T_i to the main diagonal to obtain LS A' from the LS A, for which the following formula is used: $A'[l,k] := A[l, T_i[k]], l, k = \overline{0, N-1}$. The asymptotic time complexity of the algorithm is $t \simeq O(N^2)$. Next, one needs to set the transversal T_j to the secondary diagonal, keeping the main diagonal (the elements of the main diagonal can be interchanged). After setting the transversal T_i to the main diagonal, the elements of the transversal T_j changed their position in the LS A': $T_j[k] \to T_i^{-1}[T_j[k]]$, and the values stored in them equal $v[k] = A'[k, T_i^{-1}[T_j[k]]]$. The algorithm for the targeted permutation of rows and columns of LS A' in order to obtain DLS A'' is given below.

1. Let the number of the current row $k := 0$.
2. Find column l which contains the value $v[k]$ in the k-th row.
3. Swap columns l and $N-1-k$.
4. Swap rows l and $N-1-k$.
5. Swap the values $v[k]$ and $v[N-1-k]$ in the array of values v.
6. $k := k+1$. If $k < N$, go to step 2.
7. End of the algorithm.

The algorithm sequentially processes N rows, for each of them a unary search for a suitable column is performed in linear time, therefore, the algorithm as a whole has asymptotic time complexity $t \simeq O(N^2)$.

The processing of an odd-order LS differs in that a pair of transversals placed symmetrically by Brown must have exactly one intersection $[\tilde{x}, \tilde{y}]$, the corresponding element must be set to the center of the square by rearranging rows with numbers $\tilde{x}$ and $\lfloor \frac{N}{2} \rfloor$ and columns with numbers $\tilde{y}$ and $\lfloor \frac{N}{2} \rfloor$, where $\lfloor x \rfloor$ is the operation of rounding down (truncation), $\left(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{N}{2} \rfloor\right)$ is the central cell of the square, and then set transversals T_i and T_j to both the main and secondary diagonals in the same way as discussed above.

The transversals are checked for symmetry by Brown for $\frac{|T|(|T|-1)}{2}$ pairs of transversals (upper or lower triangular submatrix with the corresponding graphic representation of the correspondence of transversals to the symmetry condition). Respectively, the asymptotic time complexity of the LS diagonalization algorithm is $t \simeq O(\frac{|T|(|T|-1)}{2}(k_1 N + k_2 N^2 + k_3 N^2)) \simeq O(|T|^2 N^2)$, where k_1, k_2, k_3 are some coefficients. In the algorithms considered above we use, as additional data structures, the mark of already considered rows and information about one of the inverse transversals T_k^{-1}, respectively, the space complexity of the algorithm is $m \simeq O(N)$. At the same time, as initial data, the algorithm operates with the initial LS A ($m \simeq O(N^2)$) and the set of its transversals ($m \simeq O(|T|N)$).

By *canonization* of a LS we mean the procedure of applying parastrophic transformations to a given LS with subsequent diagonalization of the obtained LS and, as a result, obtaining a CF of the DLS for each of the main classes of the DLS in the main class of the LS. In this case, we can restrict ourselves to using only 3 parastrophic transformations out of 6, since transposition is an equivalent transformation for the main classes of DLS and does not lead to new main classes of DLS. An example of a LS and the result of its canonization to obtain a DLS is shown in Fig. 6.

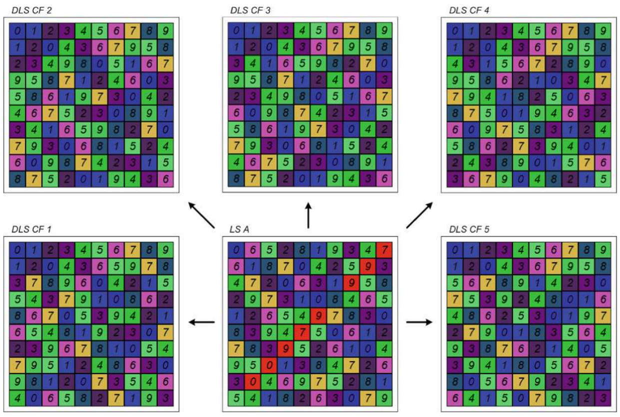

Fig. 6. An example of the original LS A and the results of its canonization (up to normalization and combinations of M-transformations): in total, LS A is diagonalized to 347 main classes of DLS, the CF of five of which are shown in the figure.

5 Practical Application of Canonization and Diagonalization

The canonization procedure can be effectively used in a random search for CF of ODLS, when from a given initial random LS, the CF is constructed from all the main classes of the DLS that are part of the corresponding main class of the LS, and then they are checked for the presence of orthogonal co-squares. The advantage in this case lies in the fact that the construction of the set of transversals is performed only once for the original LS, the sets of transversals for the DLS as part of the corresponding main classes of the DLS can be obtained from it by applying a combination of equivalent transformations (parastrophic transformations, permutations of rows and columns), which is essentially faster than building a set of transversals for each DLS separately. When constructing a list of CF of ODLS of order 10, the use of an appropriate canonizer (the application was developed by A.D. Belyshev and optimized by A.M. Albertyan [1]) makes it possible to increase the effective rate of processed DLS several-fold (see Fig. 7).

When constructing exhaustive lists of CF of ODLS for any order N (at present, the construction of the corresponding lists has been made for dimensions $N \leq 9$), the use of the canonizer does not make sense, because in this case, the initial DLS are formed by exhaustive enumeration, and the gain in the processing rate observed for random DLS will be more than compensated for by the loss either in reprocessing or filtering out of DLS that have already been processed earlier as part of the corresponding main classes of LS. In this case, the use of the classical Euler-Parker method in combination with the dancing links X algorithm (DLX) [10,11] is preferable.

For a number of LS orders, the squares with a record number of transversals are known (see the numerical sequence A090741 in OEIS[3]) [13]. By diagonalizing them, one can obtain DLS with interesting properties (for example, the maximum known number of transversals, diagonal transversals, or ODLS). Thus, a DLS with a record number of diagonal transversals for orders $N \in \{10, 12, 15\}$ and a DLS with a record number of transversals for orders $N \in \{12, 15\}$ were obtained, see Tables 1 and 2.

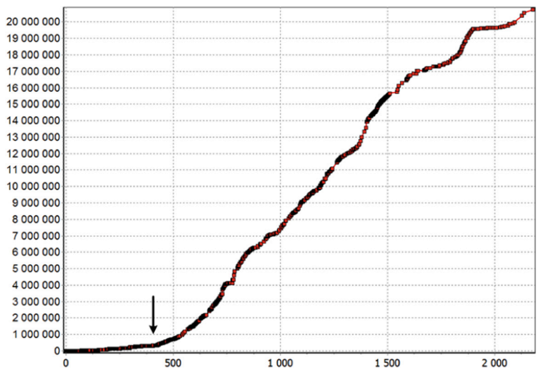

Fig. 7. The dynamics of filling the list of CF of ODLS of order 10 according to the results of calculations in the projects of volunteer distributed computing Gerasim@Home and RakeSearch: days are plotted along the X-axis, the number of CF of ODLS are along the Y-axis, the arrow marks the moment of transition from using the classical Euler-Parker method [10,11] to the canonizer.

In addition, DLS obtained as a result of diagonalization of similar LS of one of the special types (cyclic, Brown DLS, etc., depending on the order of the DLS N) [8], as a rule, form the upper part of the spectra of the corresponding numerical characteristics which seems impossible to be obtained by other methods (see example in Fig. 8).

Diagonalization of cyclic LS of order 11 yielded a number of rare combinatorial structures of ODLS of order 11, not obtained by other methods, and the highest part of the spectra of the number of diagonal transversals in DLS and ODLS of order 11 (Fig. 9).

In combination with heuristic methods for approximating the spectra S of the numerical characteristics of the DLS, based on bypassing the neighborhoods

[3] https://oeis.org/A090741.

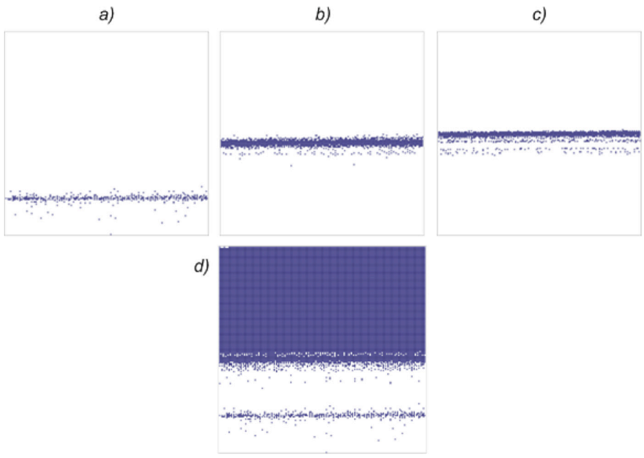

Fig. 8. Examples of spectra of the number of diagonal transversals in the DLS of order 12, obtained by diagonalizing the DLS with 198144, 132096 and 122880 transversals, respectively (a-c) and approximating the spectrum as a whole (d).

of the DLS, the use of diagonalization makes it possible both to increase the cardinality $|S|$ of the corresponding spectra and to strengthen the known upper restrictions on the lower boundary of the spectrum $\inf S$ from 43 979 to 43 093, which cannot be done by other methods.

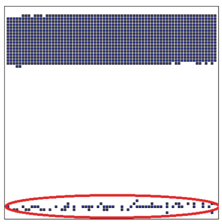

Fig. 9. Approximation of the spectrum of the number of diagonal transversals in a DLS of order 11 (the part of the spectrum obtained by diagonalizing cyclic LS is highlighted).

6 Conclusion

For DLS of order 12 with a large number of transversals and for the vast majority of DLS of order 13 and higher, a single-threaded software implementation of the diagonalization procedure takes tens of hours at best, which is why its parallel distributed software implementation was developed. Currently, with its use in one of the subprojects of the RakeSearch volunteer distributed computing project, the spectrum of the number of diagonal transversals in the DLS is being expanded to about 13. At the moment, $|S| = 12\,926$, $\inf S = 4\,756$, $\sup S = 131\,106$; about half of the experiment has been completed, the expected time of the computational experiment is about 2 months. In the perspective of further research, we plan to use diagonalization together with methods based on bypassing the DLS neighborhoods to expand the spectra of numerical characteristics of DLS of orders $N \geq 14$.

Table 1. DLS of order N with the maximum number of diagonal transversals obtained using diagonalization (integer sequence A287648 in OEIS (https://oeis.org/A287648.)).

N	DLS	Value of the numerical characteristic	Method
10	0 1 2 3 4 5 6 7 8 9 1 2 3 4 0 9 5 6 7 8 3 4 9 8 2 7 1 0 5 6 6 5 0 1 7 2 8 9 4 3 9 8 7 6 5 4 3 2 1 0 4 0 8 2 3 6 7 1 9 5 8 7 6 5 9 0 4 3 2 1 5 9 1 7 6 3 2 8 0 4 7 6 5 9 8 1 0 4 3 2 2 3 4 0 1 8 9 5 6 7	890	Extending the spectrum of the number of diagonal transversals by going around neighborhoods in combination with diagonalization
12	0 1 2 3 4 5 6 7 8 9 10 11 1 2 3 4 9 8 11 5 10 0 6 7 5 8 10 6 11 4 1 3 9 7 0 2 11 7 5 8 10 2 9 1 3 6 4 0 7 5 8 10 6 3 0 2 4 11 9 1 9 0 1 2 3 7 10 11 5 4 8 6 6 11 7 5 8 1 4 0 2 10 3 9 10 6 11 7 5 0 3 9 1 8 2 4 3 4 9 0 1 6 5 10 11 2 7 8 2 3 4 9 0 10 7 8 6 1 11 5 4 9 0 1 2 11 8 6 7 3 5 10 8 10 6 11 7 9 2 4 0 5 1 3	30 192	Diagonalization of the LS obtained by the composite squares method
15	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 1 2 0 4 5 3 7 13 9 14 11 12 10 6 8 10 11 12 13 6 7 1 2 4 5 9 14 8 0 3 13 6 7 14 8 9 4 5 12 10 0 1 2 3 11 5 3 4 10 11 12 14 8 1 2 7 13 6 9 0 12 10 11 7 13 6 0 1 3 4 8 9 14 2 5 3 4 5 11 12 10 8 9 2 0 13 6 7 14 1 6 7 13 8 9 14 5 3 10 11 1 2 0 4 12 14 8 9 1 2 0 12 10 7 13 3 4 5 11 6 4 5 3 12 10 11 9 14 0 1 6 7 13 8 2 8 9 14 2 0 1 10 11 13 6 4 5 3 12 7 2 0 1 5 3 4 13 6 14 8 12 10 11 7 9 11 12 10 6 7 13 2 0 5 3 14 8 9 1 4 7 13 6 9 14 8 3 4 11 12 2 0 1 5 10 9 14 8 0 1 2 11 12 6 7 5 3 4 10 13	4 620 434	Partial diagonalization of cyclic LS

Table 2. DLS of order N with the maximum number of transversals obtained using diagonalization (integer sequence A287644 in OEIS (https://oeis.org/A287644.)).

N	DLS	Value of the numerical characteristic	Method
12	0 1 2 3 4 5 6 7 8 9 10 11 1 2 3 4 5 0 11 6 7 8 9 10 9 8 7 6 11 10 1 0 5 4 3 2 4 5 0 1 2 3 8 9 10 11 6 7 6 11 10 9 8 7 4 3 2 1 0 5 11 10 9 8 7 6 5 4 3 2 1 0 3 4 5 0 1 2 9 10 11 6 7 8 2 3 4 5 0 1 10 11 6 7 8 9 10 9 8 7 6 11 0 5 4 3 2 1 5 0 1 2 3 4 7 8 9 10 11 6 7 6 11 10 9 8 3 2 1 0 5 4 8 7 6 11 10 9 2 1 0 5 4 3	198 144	Diagonalization of the LS obtained by the composite squares method
15	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 1 2 0 4 5 3 7 13 9 14 11 12 10 6 8 10 11 12 13 6 7 1 2 4 5 9 14 8 0 3 13 6 7 14 8 9 4 5 12 10 0 1 2 3 11 5 3 4 10 11 12 14 8 1 2 7 13 6 9 0 12 10 11 7 13 6 0 1 3 4 8 9 14 2 5 3 4 5 11 12 10 8 9 2 0 13 6 7 14 1 6 7 13 8 9 14 5 3 10 11 1 2 0 4 12 14 8 9 1 2 0 12 10 7 13 3 4 5 11 6 4 5 3 12 10 11 9 14 0 1 6 7 13 8 2 8 9 14 2 0 1 10 11 13 6 4 5 3 12 7 2 0 1 5 3 4 13 6 14 8 12 10 11 7 9 11 12 10 6 7 13 2 0 5 3 14 8 9 1 4 7 13 6 9 14 8 3 4 11 12 2 0 1 5 10 9 14 8 0 1 2 11 12 6 7 5 3 4 10 13	36 362 925	Diagonalization of cyclic LS

References

1. Albertian, A.M., Kurochkin, I.I., Vatutin, E.I.: Improving the heterogeneous computing node performance of the desktop grid when searching for orthogonal diagonal latin squares. In: Jordan, V., Tarasov, I., Faerman, V. (eds.) HPCST 2021. CCIS, vol. 1526, pp. 161–173. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94141-3_13
2. Anderson, D.P.: A platform for volunteer computing. J. Grid Comput. **18**, 99–122 (2020). https://doi.org/10.1007/s10723-019-09497-9
3. Bammel, S.E., Rothstein, J.: The number of 9 × 9 Latin squares. Discret. Math. **11**, 93–95 (1975). https://doi.org/10.1016/0097-3165(90)90015-O
4. Bedford, D.: Transversals in the Cayley tables of the non-cyclic groups of order 8. Eur. J. Comb. **12**, 455–458 (1991). https://doi.org/10.1016/S0195-6698(13)80096-0
5. Belmonte, A.R., Fiorini, E., Lenard, P., Maldonado, F., Traver, S., Wong, W.H.T.: Spectrum of transversals number in latin squares of order 1–8 (2019). https://oeis.org/A309344
6. Brown, J.W., Cherry, F., Most, L., Most, M., Parker, E.T., Wallis, W.D.: Completion of the spectrum of orthogonal diagonal Latin squares. Lect. Notes Pure Appl. Math. **139**, 43–49 (1992). https://doi.org/10.1201/9780203719916
7. Cavenagh, N.J., Wanless, I.M.: On the number of transversals in Cayley tables of cyclic groups. Disc. Appl. Math. **158**, 136–146 (2010). https://doi.org/10.1016/j.dam.2009.09.006

8. Colbourn, C.J., Dinitz, J.H.: Handbook of Combinatorial Designs, 2nd Edn. (Discrete mathematics and its applications). Chapman & Hall/CRC, Boca Raton (2006). https://doi.org/10.1201/9781420010541
9. Keedwell, A.D., Denes, J.: Latin Squares and Their Applications. Elsevier, Amsterdam (2015). https://doi.org/10.1016/C2014-0-03412-0
10. Knuth, D.E.: Dancing links (2000). arXiv preprint arXiv:cs/0011047v1
11. Knuth, D.E.: The Art of Computer Programming, vol. 4A: Combinatorial Algorithms. Addison-Wesley Professional, Boston (2013)
12. Kochemazov, S., Zaikin, O., Vatutin, E., Belyshev, A.: Enumerating diagonal Latin squares of order up to 9. J. Integer Seq. **23**, article 20.1.2 (2020)
13. McKay, B.D., McLeod, J.C., Wanless, I.M.: The number of transversals in a Latin square. Des. Codes Crypt. **40**, 269–284 (2006). https://doi.org/10.1007/s10623-006-0012-8
14. McKey, B.D., Wanless, I.M.: On the number of Latin squares. Ann. Comb. **9**, 335–344 (2005). https://doi.org/10.1007/s00026-005-0261-7
15. Sloane, N.J.A.: The on-line encyclopedia of integer sequences (2023). https://oeis.org/
16. Vatutin, E., Belyshev, A., Kochemazov, S., Zaikin, O., Nikitina, N.: Enumeration of isotopy classes of diagonal latin squares of small order using volunteer computing. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2018. CCIS, vol. 965, pp. 578–586. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05807-4_49
17. Vatutin, E.I., Belyshev, A.D., Nikitina, N.N., Manzuk, M.O.: Use of X-based diagonal fillings and ESODLS CMS schemes for enumeration of main classes of diagonal Latin squares (in Russian). Telecommunications (1), 2–16 (2023). https://doi.org/10.31044/1684-2588-2023-0-1-2-16
18. Vatutin, E.I., Kochemazov, S.E., Zaikin, O.S.: Applying volunteer and parallel computing for enumerating diagonal latin squares of order 9. In: Sokolinsky, L., Zymbler, M. (eds.) PCT 2017. CCIS, vol. 753, pp. 114–129. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67035-5_9
19. Vatutin, E.I., Kochemazov, S.E., Zaikin, O.S., Valyaev, S.Y.: Enumerating the transversals for diagonal Latin squares of small order. In: CEUR Workshop Proceedings. Proceedings of the Third International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC: FAST 2017), vol. 1973, pp. 6–14 (2017)
20. Vatutin, E.I., Titov, V.S., Zaikin, O.S., Kochemazov, S.E., Manzuk, M.O., Nikitina, N.N.: Orthogonality-based classification of diagonal Latin squares of order 10. In: CEUR Workshop Proceedings. Proceedings of the VIII International Conference "Distributed Computing and Grid-technologies in Science and Education" (GRID 2018), vol. 2267, pp. 282–287 (2018)
21. Wells, M.B.: The number of Latin squares of order 8. J. Comb. Theory **3**, 98–99 (1967). https://doi.org/10.1016/S0021-9800(67)80021-8