The Enumeration of Cyclic MNOLS

Trent G. Marbach
School of Mathematics and Physics
University of Queensland
St Lucia 4072, Australia

Fatih Demirkale
Department of Mathematics
Yildiz Technical University
Esenler, Istanbul 34220, Turkey

Diane M. Donovan
School of Mathematics and Physics
Centre for Discrete Mathematics and Computing
University of Queensland
St Lucia 4072, Australia

Abstract
In this paper we will study collections of mutually nearly orthogonal Latin squares (MNOLS), which come from a modification of the orthogonal condition for mutually orthogonal Latin squares. In particular, we find the maximum $\mu$ such that there exists a set of $\mu$ cyclic MNOLS of order $n$ for $n \leq 16$, as well as providing a full enumeration of sets and lists of $\mu$ cyclic MNOLS of order $n$ under a variety of equivalences with $n \leq 16$. This will resolve in the negative a conjecture that proposed the maximum $\mu$ for which a set of $\mu$ cyclic MNOLS of order $n$ exists is $\lceil n/4 \rceil + 1$.

Keywords: Latin square, MNOLS, nearly orthogonal.

1. Introduction
The study of mutually orthogonal Latin squares (MOLS) is a subject that has attracted much attention. Such interest has been stimulated by the relevance of the field, with applications in error correcting codes, cryptographic systems, affine planes, compiler testing, and statistics (see [8]). Although, as it is well known, there exists a set of $n - 1$ MOLS of order $n$ when $n$ is a prime or a prime power, the largest number of MOLS of order $n$ known to exist when $n$ is

Email addresses: trent.marbach@uqconenct.edu.au (Trent G. Marbach), fatihd@yildiz.edu.tr (Fatih Demirkale), dmd@maths.uq.edu.au (Diane M. Donovan)
even is generally much smaller and such sets of MOLS are hard or impossible to find; there does not exist 2 MOLS of order 6, and it is unknown whether three MOLS of order 10 exists or not.

Formally, a Latin square of order $n$ is an $n \times n$ array in which the $n$ distinct symbols $\{0, \ldots, n-1\}$ are arranged so that each symbol occurs once in each row and once in each column. We index the rows and columns by $\{0, \ldots, n-1\}$. For a Latin square $L$ we may write $(r, c, e) \in L$ to mean that $L$ has a cell in row $r$ and column $c$ that contains symbol $e$. We assume the 3 components of such a triple are taken mod $n$, so that $(r, c, e + n) = (r, c + n, e) = (r + n, c, e)$. The notation $(r, c, e) \in L$ is known as orthogonal array notation. We also write $L(r, c) = e$ when $(r, c, e) \in L$. A pair of Latin squares $L_1, L_2$ of order $n$ are called orthogonal if the superimposition of $L_1$ and $L_2$ contains each ordered pair of symbols exactly once. A set of $\mu$ Latin squares are mutually orthogonal if they are pairwise orthogonal, and we refer to such a set as a set of MOLS.

Based upon the significance and usefulness that is exhibited in the study of MOLS, Raghavarao, Shrikhande, and Shrikhande [16] introduced a modification to the definition of orthogonality to overcome restrictions for the even order case. A pair of Latin squares $L_1, L_2$ of even order $n$ are called nearly orthogonal if the superimposition of $L_1$ and $L_2$ contains each ordered pair of symbols $(l, l')$ exactly once, except in the case $l = l'$, where no such pair occurs, and in the case $l \equiv l' + n/2 \pmod{n}$, where such pairs occur twice. We consider collections of $\mu$ Latin squares of order $n$ that are pairwise nearly orthogonal, which are denoted as collections of $\mu$ MNOLS of order $n$. Traditionally these collections are unordered sets, although we will also consider ordered lists.

An orderly algorithm is a way of generating all examples of some combinatorial object, such that all equivalence classes appear in the generation, but during the generation no two objects constructed are equivalent. This technique is typically attributed to [4] and [17]. A similar technique, called canonical augmentation [10], has been used to generate Latin rectangles by augmenting a row at a time (see also [3][12]). This is not the only method of enumerating Latin rectangles, and a variety of enumerative techniques have been applied to solve it (see [13] and the citations contained within). Recently, this work has lead to the enumeration of MOLS [3]. In a similar vein, we will perform three orderly algorithms that generate collections of cyclic $\mu$ MNOLS of order $n$ under several equivalences. See [6] for a general reference on this kind of enumeration problem.

The pioneering work [16] on sets of $\mu$ MNOLS of order $n$ investigated an upper bound on $\mu$ when $n$ is fixed, and they showed that if there exists a set of $\mu$ MNOLS of order $n$, then $\mu \leq n/2 + 1$ for $n \equiv 2 \pmod{4}$ and $\mu \leq n/2$ for $n \equiv 0 \pmod{4}$. In the case that a set of $\mu$ MNOLS of order $n$ obtains this bound, it is called a complete set of $\mu$ MNOLS of order $n$.

The authors proceeded to explore the existence of sets of $\mu$ MNOLS of order $n$ by investigating $\mu$ cyclic MNOLS of order $n$; that is each Latin square $L$ has $(r, c, e) \in L$ if and only if $(r, c + 1, e + 1) \in L$, recalling that the entries are taken mod $n$. The sets of $\mu$ MNOLS of order $n$ that were found included single examples of sets of two cyclic MNOLS of order 4, three cyclic MNOLS of...
order 6, and three cyclic MNOLS of order 8, demonstrating that the bound is tight for \( n = 4 \). It was later shown \[15\] that there does not exist four MNOLS of order 6, and so the bound is not tight for \( n = 6 \).

Further results \[9\] showed sets of three MNOLS of order \( n \) exist for even \( n \geq 358 \). The authors also introduced a concept of equivalence between sets of \( \mu \) cyclic MNOLS of order \( n \) called isotopic equivalence (details in Section 2). They found the number of isotopically non-equivalent sets of \( \mu \) cyclic MNOLS of order \( n \) for \( n \leq 12 \). The number of these sets of \( \mu \) cyclic MNOLS of order \( n \) is given in Table 1.

| \( n \) | 6 | 8 | 10 | 12 |
|-------|---|---|----|----|
| \( \mu = 3 \) | 1 | 1 | ≥ 1 | > 1 |
| \( \mu = 4 \) | 0 | 0 | 1  | > 1 |
| \( \mu = 5 \) | 0 | 0 | 0  | 0  |

Table 1: The number of sets of \( \mu \) MNOLS of order \( n \) under isotopic equivalence.

The existence of three cyclic MNOLS of orders \( 48k+14, 48k+22, 48k+38 \), and \( 48k+46 \) when \( k \geq 0 \) was documented in \[2\], and also in \[1\], which verified that there exists a set of three MNOLS of order \( n \) for all even \( n \geq 6 \), except perhaps when \( n = 146 \).

In the current paper, we find the maximum \( \mu \) such that there exists a set of \( \mu \) cyclic MNOLS of order \( n \) for \( n \leq 16 \), as well as providing a full enumeration of sets and lists of \( \mu \) cyclic MNOLS of order \( n \) under a variety of equivalences with \( n \leq 16 \). This will resolve in the negative a conjecture of \[9\] that proposed the maximum \( \mu \) for which a set of \( \mu \) cyclic MNOLS of order \( n \) exists is \( \lceil n/4 \rceil + 1 \) (the maximum \( \mu \) appears erroneously as \( \lceil n/8 \rceil + 1 \) in the original conjecture \[18\], and the maximum value we have written was the intended conjecture).

2. Further definitions

We will enumerate both ordered lists and sets of \( \mu \) MNOLS of order \( n \). A set of \( \mu \) MNOLS of order \( n \) is a set \( \{L_1, \ldots, L_\mu\} \) such that \( L_i, L_j \) are nearly orthogonal for \( 1 \leq i, j \leq \mu, i \neq j \). A list of \( \mu \) MNOLS of order \( n \) is an ordered list \( (L_1, \ldots, L_\mu) \) such that \( L_i, L_j \) are nearly orthogonal for \( 1 \leq i, j \leq \mu, i \neq j \). This distinction will be important when we enumerate collections of \( \mu \) MNOLS of order \( n \). A list of \( \mu \) MNOLS of order \( n \) \( (L_1, \ldots, L_\mu) \) is reduced if \( L_1 \) has its first row and column in natural order.

Define an ordering \( \prec \) on the set of all Latin squares of order \( n \) as follows: for two Latin squares, \( L \) and \( M \), we have \( L \prec M \) if and only if either \( L = M \) or there is some \( i, j \in \{1, \ldots, n\} \) such that \( L(i, j) < M(i, j) \) and \( L(i', j') = M(i', j') \) whenever either \( i' < i \) or both \( i' = i \) and \( j' < j \). It can be proved that this is a total ordering.

Take the total order \( \bowtie \) on lists on \( \mu \) MNOLS of order \( n \) to be such that for two lists of \( \mu \) MNOLS of order \( n \), \( L = (L_1, \ldots, L_n) \) and \( M = (M_1, \ldots, M_n) \),
we have $L \triangleleft M$ if either $L = M$ or there is $1 \leq \beta \leq \mu$ such that $L_\beta \triangleleft M_\beta$ and $L_{\beta'} = M_{\beta'}$ for $\beta' < \beta$.

Given a collection of Latin squares, $\mathcal{L}$, and $(\sigma_R, \sigma_C, \sigma_E) \in S_3^3$, we define $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ to be the collection of Latin squares that is obtained by uniformly permuting the rows (respectively columns, symbols) of the Latin squares in $\mathcal{L}$ by $\sigma_R$ (respectively $\sigma_C$, $\sigma_E$). Then each $(\sigma_R, \sigma_C, \sigma_E) \in S_3^3$ is a group action, acting on the set of collections of Latin squares. Two lists of $\mu$ $MNOLS$ of order $n$, $\mathcal{L}$ and $\mathcal{N}$, are list-isotopic if there exists $(\sigma_R, \sigma_C, \sigma_E) \in S_3^3$ with $\mathcal{N} = \mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$. Two sets of $\mu$ $MNOLS$ of order $n$ are isotopic if some ordering of the Latin squares within each set gives two list-isotopic lists of $\mu$ $MNOLS$ of order $n$. Two lists of $\mu$ $MNOLS$ of order $n$, $\mathcal{L}$ and $\mathcal{N}$, are set-isotopic if forgetting the order on the lists gives two isotopic sets of $\mu$ $MNOLS$ of order $n$. Clearly the number of sets of $\mu$ $MNOLS$ of order $n$ that are distinct up to isotopy is the same as the number of lists of $\mu$ $MNOLS$ of order $n$ up to set-isotopy.

3. Cyclic MNOLS

We take $I = I_n \in S_n$ to be the identity permutation of order $n$, $\tau = \tau_n \in S_n$ to be the permutation that is a cyclic shift of size one, i.e. $\tau(j) \equiv j + 1 \pmod{n}$ for $0 \leq j < n$, and $m_x = m_{x,n} \in S_n$ to be the permutation defined as $m_x(j) = j \cdot x \pmod{n}$ for $0 \leq j < n$, where $1 \leq x < n$ and $\gcd(x,n) = 1$. In the following four lemmata, we endeavor to describe properties of cyclic $MNOLS$ when they are acted upon by the group actions $(\sigma_R, \sigma_C, \sigma_E) \in S_3^3$. For the following, recall that each component of a triple $(r, c, e) \in L$ is taken modulo $n$.

Taking a collection of $\mu$ $MNOLS$ of order $n$, $\mathcal{L}$, if $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ is a collection of $\mu$ $MNOLS$ of order $n$ then it is immediate clear that $\sigma_E(y + n/2) = \sigma_E(y) + n/2$ for all $0 \leq y < n/2$, in order to preserve near orthogonality. The converse is also true:

Lemma 3.1. Consider a collection of $\mu$ $MNOLS$ of order $n$, $\mathcal{L}$, along with $(\sigma_R, \sigma_C, \sigma_E) \in S_3^3$ such that $\sigma_E(y + n/2) \equiv \sigma_E(y) + n/2 \pmod{n}$ for all $0 \leq y < n/2$. Then $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ is also a collection of $\mu$ $MNOLS$ of order $n$.

Proof. Take Latin squares $L_1 \neq L_2$ to be in $\mathcal{L}$ and Latin squares $L_1', L_2'$ to be $L_1$ and $L_2$ with rows, columns, and symbols permuted by $\sigma_R$, $\sigma_C$, and $\sigma_E$. Consider the multisets $P = \{(e, e') \mid (r, c, e) \in L_1$ and $(r, c, e') \in L_2$ where $r, c \in \{0, \ldots, n - 1\}\}$ and also $P' = \{(e, e') \mid (r, c, e) \in L_1'$ and $(r, c, e') \in L_2'$ where $r, c \in \{0, \ldots, n - 1\}\}$. The ordered pair $(e, e')$ appears in $P$ if and only if the ordered pair $(\sigma_E(e), \sigma_E(e'))$ appears in $P'$. As each pair $(e, e + n/2)$ appear twice in $P$ (recalling that each symbol is taken modulo $n$), then each pair $(\sigma_E(e), \sigma_E(e' + n/2)) = (\sigma_E(e), \sigma_E(e + n/2))$ appear twice in $P'$. No pair $(e, e)$ appears in $P'$, or else $(\sigma_E^{-1}(e), \sigma_E^{-1}(e))$ would appear in $P$, which we know it does not. Each other possibility $(e, e')$ clearly appears precisely once in $P'$ because $(\sigma_E^{-1}(e), \sigma_E^{-1}(e'))$ appeared precisely once in $P$. Therefore the conditions are satisfied for $L_1'$ and $L_2'$ to be nearly orthogonal to each other, and so $\mathcal{L}'$ is a collection of $\mu$ $MNOLS$ of order $n$.
Permuting the rows in a collection of cyclic MNOLS does not destroy the cyclic property:

**Lemma 3.2.** Consider a collection of $\mu$ MNOLS of order $n$, $L$, and permutation $\sigma_R \in S_n$. If $L$ is cyclic, then so is $L(\sigma_R, I, I)$.

*Proof.* Consider Latin squares $L \in \mathcal{L}$ and $L' \in \mathcal{L}(\sigma_R, I, I)$ such that $L'$ is the Latin square that is obtained by permuting the row indexes of $L$ by $\sigma$. For every cell $(r, c, e) \in L'$, there must be a cell $(\sigma^{-1}(r), c, e) \in L$. As $L$ is cyclic, $(\sigma^{-1}(r), c + 1, e + 1) \in L$, but this must mean that $(\sigma(\sigma^{-1}(r)), c + 1, e + 1) = (r, c + 1, e + 1) \in L'$. This implies $L(\sigma_R, I, I)$ is cyclic.

Simultaneously cycling the columns and symbols has no effect on a collection of cyclic MNOLS:

**Lemma 3.3.** Consider a collection of $\mu$ cyclic MNOLS of order $n$, $L$. Then $L = \mathcal{L}(I, \tau^i, \tau^i)$, for $0 \leq i < n$, recalling $\tau(x) = x + 1 \mod n$ for $x \in \{0, \ldots, n-1\}$.

*Proof.* Consider Latin squares $L \in \mathcal{L}$ and $L' \in \mathcal{L}(I, \tau^i, \tau^i)$ such that $L'$ is obtained by permuting the column and symbol indexes of $L$ by $\tau^i$. As $L$ is cyclic, for every cell $(r, c, e) \in L$, there must be a cell $(r, c - i, e - i) \in L$. But then $(r, \tau^i(c - i), \tau^i(e - i)) = (r, c, e) \in L'$. The result follows.

If applying some permutation to the columns and to the symbols of a collection of MNOLS does not remove the cyclic property, then the permutation applied must be of a certain form:

**Lemma 3.4.** Consider a collection of $\mu$ MNOLS of order $n$, $L$, and permutations $\sigma_C, \sigma_E \in S_n$ such that $\sigma_E(y + n/2) \equiv \sigma_E(y) + n/2 \mod n$, for all $0 \leq y < n/2$. If both $L$ and $\mathcal{L}(I, \sigma_C, \sigma_E)$ are cyclic, then $\sigma_C = m \cdot \tau^\sigma_C(0)$ and $\sigma_E = m \cdot \tau^\sigma_E(0)$, where $x = \sigma_E(1) - \sigma_E(0) = \sigma_C(1) - \sigma_C(0)$ and $\gcd(x, n) = 1$.

*Proof.* Take $L \in \mathcal{L}$ and $L' \in \mathcal{L}(I, \sigma_C, \sigma_E)$, so that $(r', c', e') \in L$ if and only if $(r', \sigma_C(c'), \sigma_E(e')) \in L'$. For each symbol $i$ there is a row $r$ with $(r, 0, i) \in L$, and because $L$ is cyclic $(r, 1, i + 1) \in L$. Then $(r, \sigma_C(0), \sigma_E(i)), (r, \sigma_C(1), \sigma_E(i + 1)) \in L'$, and because $L'$ is cyclic, $\sigma_E(i + 1) - \sigma_E(i) = \sigma_C(1) - \sigma_C(0)$. Thus $\sigma_E(i + 1) - \sigma_E(i)$ is independent of $i$ and so set $\sigma_E(i + 1) - \sigma_E(i) = x$. This gives $\sigma_E(1) - \sigma_E(0) = \sigma_C(1) - \sigma_C(0)$ when we take $i = 0$. This also gives $\sigma_E(i) = \sigma_E(0) + i \cdot x$, or equivalently $\sigma_E = m \cdot \tau^\sigma_E(0)$. Since $\sigma_E$ is a permutation, we must have $\gcd(x, n) = 1$.

For each column $j$ there is a row $r$ with $(r, j, 0) \in L$, and because $L$ is cyclic $(r, j + 1, 1) \in L$. Then $(r, \sigma_C(j), \sigma_E(0)), (r, \sigma_C(j + 1), \sigma_E(1)) \in L'$, and because $L'$ is cyclic, $\sigma_C(j + 1) - \sigma_C(j) = \sigma_E(1) - \sigma_E(0) = x$ for all $0 \leq i < n$. This gives $\sigma_C(j) = \sigma_C(0) + j \cdot x$, or alternately $\sigma_C = m \cdot \tau^\sigma_C(0)$. 

\[\square\]
These facts come together to tell us exactly the form of any non-trivial group actions that preserve the cyclic property when applied to a list of MNOLS:

**Lemma 3.5.** Consider collection of \( \mu \) cyclic MNOLS of order \( n \), \( L \) and \( N \). Then \( N = \mathcal{L}(\sigma_R, \sigma_C, \sigma_E) \) if and only if there exists integers \( 0 \leq x, j' < n \) with \( \gcd(x, n) = 1 \), such that \( N = \mathcal{L}(\sigma_R, m_x, m_x \cdot \tau'^j) \).

**Proof.** The reverse implication is clear, so we show the forward implication. Because \( N = \mathcal{L}(\sigma_R, \sigma_C, \sigma_E) \) is cyclic, then \( \mathcal{L}(I, \sigma_C, \sigma_E) \) is cyclic by Lemma 3.2, which by Lemma 3.4 implies \( \sigma_C = m_x \cdot \tau^x \) and \( \sigma_E = m_x \cdot \tau^{\sigma E(0)} \) for some \( x \) with \( \gcd(x, n) = 1 \). But then \( N = \mathcal{L}(\sigma_R, m_x \cdot \tau^{\sigma E(0)}, m_x \cdot \tau^{\sigma E(0)}) = \mathcal{L}(\sigma_R, m_x, m_x \cdot \tau^{\sigma E(0)-\sigma E(0)}) \) by Lemma 3.3 with \( i = -\sigma C(0) \). Taking \( j' = \sigma E(0) - \sigma C(0) \) yields the result. \( \square \)

This aligns with previous work of Li and van Rees \([9]\) on isomorphisms of \((t,2m)\)-difference sets, which are equivalent to set-isotopisms of \( t \) MNOLS of order \( 2m \).

4. Group actions

Let \( C_n^\mu \) be the set of all lists of \( \mu \) cyclic MNOLS of order \( n \). From the previous section, not every action of \( S_n^\mu \) applied to an element of \( C_n^\mu \) yields an element of \( C_n^\mu \). In fact, Lemma 3.5 informs us of the exact actions that can be used. Given a list of \( \mu \) cyclic MNOLS of order \( n \), \( L = (L_1, \ldots, L_\mu) \), define the actions \( \mathfrak{m}_x(L) = L(I, m_x, m_x), \mathfrak{T}(L) = L(I, I, \tau), \) and \( r_\sigma(L) = L(\sigma, I, I) \), for \( \sigma \in S_n, 1 \leq x \leq n-1, \) and \( \gcd(x, n) = 1 \). Then the actions of \( S_n^\mu \) that map lists of \( \mu \) cyclic MNOLS of order \( n \) to lists of \( \mu \) cyclic MNOLS of order \( n \) are those that can be decomposed into an action \( \mathfrak{m}_x \cdot \mathfrak{T} \cdot r_\sigma \), for \( \sigma \in S_n, 1 \leq j, x \leq n, \) and \( \gcd(x, n) = 1 \).

Given a list of \( \mu \) cyclic MNOLS of order \( n \), \( L = (L_1, \ldots, L_\mu) \), we define the actions \( s_\sigma(L) = (L_{\sigma(1)}, \ldots, L_{\sigma(\mu)}) \), for \( \sigma \in S_n \). The orbit of these actions on \( L \) contains as its elements exactly those lists of \( \mu \) MNOLS of order \( n \) that become the set \( \{L_1, \ldots, L_\mu\} \) when we forget the ordering on the list.

Define \( M = \{m_x | 1 \leq x \leq n-1 \text{ and } \gcd(x, n) = 1 \} \), \( T = \{t^i | 0 \leq i \leq n-1 \} \), \( R = \{r_\sigma | \sigma \in S_n \} \), and \( S = \{s_\sigma | \sigma \in S_n\} \), each of which form a group under composition.

We have previously defined a reduced list of \( \mu \) MNOLS of order \( n \), but we are yet to define a reduced set of \( \mu \) MNOLS of order \( n \). Take \( L \) to be a list of \( \mu \) MNOLS of order \( n \). Notice that the least element under \( \triangleleft \) in the orbit \( R \cdot L \) is a reduced list of \( \mu \) MNOLS of order \( n \). We call the least element, say \( (L_1, \ldots, L_\mu) \), under \( \triangleleft \) in the orbit \( (R, S) \cdot L \) set-reduced, and further we call the set \( \{L_1, \ldots, L_\mu\} \) reduced.

We consider the actions of the group \( \langle M, T, R, S \rangle \) on \( C_n^\mu \), which has order \( \phi(n) \cdot n \cdot n! \cdot \mu! \). The orbits of this action are the set-isotopy classes of \( C_n^\mu \), and two lists in the same set-isotopy class are set-isotopic. The stabilizer \( I_{s_\alpha}(L) = \{\alpha \in \langle M, T, R, S \rangle | \alpha(L) = L\} \) is called the set-autotopy group, and each contained
element is called a set-autoptpy. The number of distinct orbits under this group is the number of sets of \( \mu \) MNOLS of order \( n \) distinct up to isotopy.

We will also be interested in the subgroups \( \langle M, T, R \rangle \leq \langle M, T, R, S \rangle, \langle R, S \rangle \leq \langle M, T, R, S \rangle, R \leq \langle M, T, R, S \rangle, \text{ and } S \leq \langle M, T, R, S \rangle \). Their orbits are called, respectively, the list-isotopy classes, set-reduced classes, the list-reduced classes, and the set classes of \( C_n^\mu \). We call the stabilizers of \( L \in C_n^\mu \), respectively, \( Is_l(L) \), \( Red_s(L) \), \( Red_l(L) \), and \( Set(L) \). The number of distinct orbits under these subgroups are respectively the number of lists distinct up to isotopy, the number of reduced sets, the number of reduced lists, and the number of sets of \( \mu \) MNOLS of order \( n \).

This paper will find for \( n \leq 16 \) and for each \( 2 \leq \mu \leq 5 \):

1. the number of isotopy classes of sets of \( \mu \) cyclic MNOLS of order \( n \);
2. the number of isotopy classes of lists of \( \mu \) cyclic MNOLS of order \( n \);
3. the number of reduced sets of \( \mu \) cyclic MNOLS of order \( n \);
4. the number of reduced lists of \( \mu \) cyclic MNOLS of order \( n \);
5. the number of sets of \( \mu \) cyclic MNOLS of order \( n \);
6. the number of lists of \( \mu \) cyclic MNOLS of order \( n \);

The identity action is the only action in \( \langle R \rangle \) and \( \langle S \rangle \) that is a stabilizer, hence \( |Red_l(L)| = 1 \) and \( |Set(L)| = 1 \), the orbits of \( \langle R \rangle \) have constant size \( n! \), and the orbits of \( \langle S \rangle \) have constant size \( \mu! \). Each set-isotopy class can be divided into classes corresponding to the other subgroups, and is closed in the sense that two lists or sets that are equivalent under any of the equivalences must appear in the same set-isotopy class. Our primary approach for the computer search for this problem is to find a list of \( \mu \) MNOLS of order \( n \) that represents each set-isotopy classes, and count how many classes of each type appear within this set-isotopy class:

**Lemma 4.1.** Given \( L \in C_n^\mu \):

1. the number of list-isotopy classes within the set-isotopy class of \( L \) is:
   \[ \mu! \cdot \frac{|Is_l(L)|}{|Is_s(L)|}; \]
2. the number of set-reduced classes within the set-isotopy class of \( L \) is:
   \[ \phi(n) \cdot n \cdot \frac{|Red_s(L)|}{|Is_s(L)|}; \]
3. the number of list-reduced classes within the set-isotopy class of \( L \) is:
   \[ \phi(n) \cdot n \cdot \mu! \cdot \frac{|Red_l(L)|}{|Is_s(L)|} = \phi(n) \cdot n \cdot \mu!/|Is_s(L)|; \]
4. the number of set classes within the set-isotopy class of \( L \) is:
   \[ \phi(n) \cdot n \cdot n! \cdot \frac{|Set(L)|}{|Is_s(L)|} = \phi(n) \cdot n \cdot n!/|Is_s(L)|; \text{ and} \]
5. the number of lists within the set-isotopy class of \( L \) is:
   \[ \phi(n) \cdot n \cdot \mu! \cdot n!/|Is_s(L)|. \]

**Proof.** By the orbit-stabilizer theorem. \( \square \)

## 5. Canonical forms

Given a partition of \( C_n^\mu \) as \( C_n^\mu = \cup_{i=1}^{\alpha} C_i \) with \( C_i \cap C_j = \emptyset \) for \( 1 \leq i < j \leq \alpha \), a **canonical form** is a function \( f : C_n^\mu \to C_n^\mu \) such that for all \( \mathcal{L}, \mathcal{M} \in C_i \),
We will say the lists within $\text{Im}(f)$ are canonical. We call $M \in C_i$ with $M = f(M)$ the canonical representation of $C_i$ (each of these $M$ are canonical). This allows us to represent each orbit of the elements of a group acting on $C_n^\mu$ by a single list of $\mu$ MNOLS of order $n$.

Typically, enumerations involving Latin squares \cite{5,13} utilize a conversion from a Latin square to a labeled graph where an isotopism applied to the Latin square corresponds to a certain relabeling of the vertices of the graph. Programs such as nauty \cite{11} can be used to find a canonical labelling of a graph, and by comparing the canonical labeling of two graphs that correspond to two Latin squares, it is easy to evaluate whether the two Latin squares are isotopic or not. Such programs are useful because they are the fastest implemented solutions for the graph isomorphism problem, which is in $NP$, and hence also for determining when two Latin squares are isotopic or not.

During initial investigations it was found that the greatest portion of time taken by our enumeration programs was spent calculating which pairs of Latin squares were nearly orthogonal. Of these pairs of Latin squares, only a small portion were nearly orthogonal and would then go on to be checked for isotopisms, so isotopism checking occurred relatively infrequent compared to checking for nearly orthogonality. This is atypical for Latin square based enumeration problems, as it is common that the greatest amount of time is spent on isotopism checking. So while external software to find canonical representations of classes of MNOLS may very well speed up our program, this speed up will be negligible. Due to this, we have opted to use an explicit canonical form that is easy to explain.

We will use a canonical form with the property that removing the last Latin square of any list of $\mu$ MNOLS of order $n$ that is canonical yields a list of $\mu - 1$ MNOLS of order $n$ that is canonical. Then, as will be in two of our three algorithms, our approach is to use all lists of $\mu - 1$ MNOLS of order $n$ that are canonical, and extend these to lists of $\mu$ MNOLS of order $n$ that are canonical. We also wish to have some knowledge about which Latin squares we can append in order to avoid, as much as possible, creating a list of $\mu$ MNOLS of order $n$ that are not canonical.

Let $M_i \in C_i$ be such that $M_i \triangleleft \mathcal{L}$ for each $\mathcal{L} \in C_i$. In this paper we will consider the canonical form $f$ defined by $f(\mathcal{L}) = M_i$, for $\mathcal{L} \in C_i$. For example if we take $C_i$ to be the orbits of $\langle R \rangle$, the canonical representation of each orbit is the unique list of $\mu$ MNOLS of order $n$ amongst the orbit that is reduced.

This aligns with our previous definition of reduced lists of $\mu$ MNOLS of order $n$, so rather that saying such a list is canonical, we will now say it is list-reduced. If instead we take $C_i$ to be the orbits of $\langle R, S \rangle$, we call the canonical representation of each orbit set-reduced. Similarly we say a list is set-canonical if we take the partition of $C_n^\mu$ into set-isotopy classes, and list-canonical if we take the partition of $C_n^\mu$ into list-isotopy classes.

**Lemma 5.1.** Given the partition of $C_n^\mu$ into set-isotopy classes and $f$ a canonical form:

1. The number of set-isotopy classes in $C_n^\mu$ is $|\text{Im}(f)|$. 

8
2. The number of list-isotopy classes in $C_n^\mu$ is $\sum_{\mathcal{L} \in \text{Im}(f)} \mu! \cdot |I_{\text{Sl}}(\mathcal{L})|/|I_{\text{Sl}}(\mathcal{L})|$.  
3. The number of set-reduced classes in $C_n^\mu$ is $\sum_{\mathcal{L} \in \text{Im}(f)} |\phi(n)| \cdot n! / |I_{\text{Sl}}(\mathcal{L})|$.  
4. The number of list-reduced classes in $C_n^\mu$ is $LR = \sum_{\mathcal{L} \in \text{Im}(f)} |\phi(n)| \cdot n! \cdot \mu! / |I_{\text{Sl}}(\mathcal{L})|$.  
5. The number of set classes in $C_n^\mu$ is $\sum_{\mathcal{L} \in \text{Im}(f)} |\phi(n)| \cdot n! / |I_{\text{Sl}}(\mathcal{L})| = LR \cdot n!$.  
6. The number of lists in $C_n^\mu$ is $\sum_{\mathcal{L} \in \text{Im}(f)} |\phi(n)| \cdot n! \cdot \mu! / |I_{\text{Sl}}(\mathcal{L})| = LR \cdot n!$.

Proof. A consequence of Lemma 4.1. \qed

6. Algorithm

There has been a history of errors in the enumeration of Latin squares (this history is described in [14]). As such, it has become standard practice in the enumeration of Latin squares and related structures to run at least two distinct programs to enumerate using two different methods, and check the results are identical. We present 3 different algorithms that when implemented independently by two authors arrived at the same results for $\mu \geq 2$ and $n \leq 14$. One of these algorithms when implemented by one author found a result for $\mu \geq 2$ and $n = 16$.

Any cyclic Latin square can be generated by its first column. As such, we look for lists of $\mu$ columns of size $n$ that can generate lists of $\mu$ cyclic MNOLS of order $n$.

Algorithm A constructs all list-reduced lists of $\overline{n}$ cyclic MNOLS of order $n$. Algorithm B and Algorithm C will only construct the set-canonical representations of each set-isotopy class, with the difference being that Algorithm B uses a basic depth first search, while Algorithm C saves information of those cyclic Latin squares that may be used, and uses that to reduce repeated calculations.

Algorithm A works by using the canonical representation of a list-reduced class of $\overline{n}$ cyclic MNOLS of order $n$, and adding possible cyclic Latin squares in order to yield lists of $\overline{n}$ cyclic MNOLS of order $n$ that are list-reduced, where $2 \leq \overline{n} \leq \mu$. In the case the result is set-canonical, we calculate $|\text{Red}_s(\mathcal{L})|$, $|\text{Sl}(\mathcal{L})|$, and $|I_{\text{Sl}}(\mathcal{L})|$. Whether or not the result was set-canonical, we continue to add more cyclic Latin squares recursively. After completion, we merge the results and use Lemma 6.1 to find the total number of classes.

Algorithm B works by using the set-canonical representation of a set-isotopy class of $\overline{n}$ cyclic MNOLS of order $n$, and adding possible cyclic Latin squares in order to yield lists of $\overline{n}$ cyclic MNOLS of order $n$ before checking whether the resulting lists are set-canonical, where $2 \leq \overline{n} \leq \mu$. If a result, say $\mathcal{L}$, is set-canonical, it is the canonical representation of a set-isotopy class of $\overline{n}$ cyclic MNOLS of order $n$. In such a case, we calculate $|\text{Red}_s(\mathcal{L})|$, $|\text{Sl}(\mathcal{L})|$, and $|I_{\text{Sl}}(\mathcal{L})|$, before continuing to add more cyclic Latin squares recursively. After completion, we merge the results and use Lemma 6.1 to find the total number of classes.

Algorithm C begins by generating all columns that could be used to generate the second Latin square in a list of list-reduced two MNOLS of order $n$ (See
Algorithm 1: Algorithm A

input: An integer $\mu$ with $\mu < \mu$;
A list of $\mu$ columns $(C_1, \ldots, C_\mu)$ that generate a list of $\mu$ cyclic MNOLS of order $n$ that is list-reduced;

output: The quadruple of integers ((1),(2),(3),(4)), a count of the number of lists of $\mu$ columns ($C_1, \ldots, C_\mu, D_{\mu+1}, \ldots, D_\mu$) that generate lists of $\mu$ cyclic MNOLS of order $n$ that are (1) list-reduced, (2) set-reduced, (3) list-canonical, and (4) set-canonical;

1. $sum \leftarrow (0, 0, 0, 0)$;
2. function $extend1(\mu, C_1, \ldots, C_\mu)$
3. for columns $P$ that form a permutation do
4.  
5.   for $i = 1 : \mu$ do
6.     if The pair of columns $(C_i, P)$ does not generate two cyclic MNOLS of order $n$ then
7.       go to the next possible column.
8.     $L \leftarrow$ the list of $\mu$ cyclic MNOLS of order $n$ generated by $(C_1, \ldots, C_\mu, P)$
9.     if $L = \mu - 1$ then
10.       if $L$ is set-canonical then
11.          $sum \leftarrow sum + (1, 1, 1, 1)$
12.       else
13.          if $L$ is list-canonical then
14.             $sum \leftarrow sum + (1, 1, 1, 0)$
15.             else
16.                 if $L$ is set-reduced then
17.                     $sum \leftarrow sum + (1, 1, 0, 0)$
18.                 else
19.                   $sum \leftarrow sum + (1, 0, 0, 0)$
20.       else
21.         $sum \leftarrow sum + extend1(\mu + 1, C_1, \ldots, C_\mu, P)$;
22. return $sum$;
## Algorithm 2: Algorithm B

**Input:** An integer \( \mu \) with \( \mu < \mu' \);  
A list of \( \mu' \) columns \((C_1, \ldots, C_{\mu'})\) that generate a list of cyclic \( \mu' \) \textit{MNOLS} of order \( n \) that is set-canonical;  

**Output:** A multiset of ordered triples, \( \text{store} \), that for each set-isotopy class of \( \mu \) \textit{MNOLS} of order \( n \) with canonical representative \( \mathcal{L} \) that can be generated from \((C_1, \ldots, C_{\mu'}, D_{\mu'+1}, \ldots, D_{\mu})\), \( \text{store} \) contains one triple \((\text{Is}_s(\mathcal{L}), \text{Is}_l(\mathcal{L}), \text{Red}_s(\mathcal{L}))\);  

1. \( \text{store} \leftarrow \emptyset \);  
2. **procedure** \( \text{extend2}(\nu, C_1, \ldots, C_{\nu'}) \)  
3. for columns \( P \) that form a permutation do  
4.  
5.  
6.  
7. \( \mathcal{L} \leftarrow \) the list of \( \nu + 1 \) cyclic \textit{MNOLS} of order \( n \) generated by \((C_1, \ldots, C_{\nu'}, P)\)  
8. if \( \mathcal{L} \) is set-canonical then  
9.  
10.  
11. else  
12. \( \text{store} \leftarrow \text{store} \cup \text{extend}(\nu + 1, C_1, \ldots, C_{\nu'}, P) \)  
13. return \( \text{store} \)
Algorithm A). It then places those columns that form set-canonical MNOLS in list1, those that do not have 1 as their first entry in list2, and throwing away those that have 1 as their first entry but do not form set-canonical MNOLS. For each \( A \in \text{list1} \), create a list list3 that contains each \( B \in \text{list2} \) such that \((A, B)\) generates a list of two cyclic MNOLS of order \( n \). Construct a graph with vertices in list3, and edges connecting points \( B_1 \) and \( B_2 \) if \((B_1, B_2)\) generates a list of two cyclic MNOLS of order \( n \). Then each clique \((e_1, \ldots, e_\alpha)\) corresponds to \((\alpha + 2)\) cyclic MNOLS of order \( n \), generated by \((I, A, e_1, \ldots, e_\alpha)\). For each clique, if the generated \((\alpha + 2)\) cyclic MNOLS of order \( n \), \( L \), is set-canonical, we calculate \(|\text{Red}_s(L)|\), \(|\text{Is}_s(L)|\), and \(|\text{Is}_s(L)|\). After completion, we merge the results and use Lemma 5.1 to find the total number of classes.

Finding cliques is usually a hard problem. This is not an issue for our calculations as the clique size of our problem turns out to be very small. In fact, no cliques of size three existed in our graph, and the computation time to prove this was negligible within our program as a whole.

7. Results and conclusions

The counts that were found appear in Tables 2, 3, 4, and 5. Comparing these results to the previously known cases in Table 1 we see that the new values of particular significance are when \( \mu = 3 \) and \( n \in \{10, 12, 14, 16\} \), when \( \mu = 4 \) and \( n \in \{12, 14, 16\} \), and when \( \mu = 5 \) and \( n \in \{14, 16\} \). The results when \( \mu = 5 \) disproves Conjecture 5.2 of [8] that proposed the maximum \( \mu \) for which a set of \( \mu \) cyclic MNOLS of order \( n \) exists is \([n/4] + 1\), as there does not exist five MNOLS of order 14 and five MNOLS of order 16 as predicted by the conjecture.

For \( n = 14 \), the search using Algorithm A consumed 372.7 days of CPU time, using Algorithm B consumed 19,695 hours of CPU time, and using Algorithm C consumed 3,956 hours of CPU time. Algorithm C consumes a great deal more memory than the other methods. We ran Algorithm C for \( n = 16 \), which consumed 154.05 days of CPU time and over 7GBs of RAM was required. This method therefore would have to be significantly modified to reduce the memory usage if an attempt was made for running it with parameters \( n = 18 \).

We say a list of \( \mu \) MNOLS of order \( n \) is of type 0 if it is isotopically equivalent to a list of reduced \( \mu \) MNOLS of order \( n \), \( L = (L_1, \ldots, L_\mu) \), with \((0, 0, 1), (1, 0, 0) \in L_2 \), and is of type 1 otherwise. A set of \( \mu \) MNOLS of order \( n \) is of type 0 if fixing the order in some way gives a list of \( \mu \) MNOLS of order \( n \) of type 0, and is of type 1 otherwise.

A collection of \( \mu \) cyclic MNOLS of order \( n \) contains a row-intercalate of difference \( d \) if two of its Latin squares \( L \) and \( M \) have two rows \( r, r' \) with \( r < r' \) and \( r' - r = d \) such that \((r, 0, e) \in L \) if and only if \((r', 0, e) \in M \), and also \((r', 0, e') \in L \) if and only if \((r, 0, e') \in M \), for some \( e, e' \in \{0, \ldots, n - 1\} \). Then it is clear that a collection of \( \mu \) MNOLS of order \( n \) is of type 0 if and only if it contains a row-intercalate of difference \( d \) and \( \gcd(d, n) = 1 \). Clearly set-isotopy preserves type. In Tables 9, 10, and 11 we show the number of set-isotopy classes of each type. Observe that the proportion of set-isotopy classes
Algorithm 3: Algorithm C

**input**: 1/ A list, list1, that contains all lists of columns such that
\( C \in \text{list1} \) implies the pair of columns \((I, C)\) generates a list of
two cyclic MNOLS of order \( n \) that is set-canonical;
2/ A list, list2, that contains all lists of columns such that
\( C \in \text{list2} \) implies the pair of columns \((I, C)\) generates a list of
two cyclic MNOLS of order \( n \) that is list-reduced and \( C \) does
not contain 1 as its first element;

**output**: A multiset of ordered triples, store, that for each set-isotopy
class of \( \mu \) cyclic MNOLS of order \( n \) with canonical
representative \( \mathcal{L} \), store contains one triple
\((\text{Is}_s(\mathcal{L}), \text{Is}_l(\mathcal{L}), \text{Red}_s(\mathcal{L}))\);

1. \( \text{store} \leftarrow \emptyset; \)
2. \( \text{vert}(\text{graph}) \leftarrow \emptyset; \)
3. \( \text{edge}(\text{graph}) \leftarrow \emptyset; \)
4. for \( C_1 \in \text{list1} \) do
5. \( \text{list3} \leftarrow \emptyset; \)
6. for \( C_2 \in \text{list2} \) do
7. \( \text{if} \ (C_1, C_2) \) generates a list of two cyclic MNOLS of order \( n \) then
8. \( \text{list3} \leftarrow \text{list3} \cup C_2 \)
9. \( \text{vert}(\text{graph}) \leftarrow \text{list3} \)
10. for \( C_3 \in \text{list3} \) do
11. \( \text{for} \ C_4 \in \text{list3} \) do
12. \( \text{if} \ (C_3, C_4) \) generates a list of two cyclic MNOLS of order \( n \) then
13. \( \text{edge}(\text{graph}) \leftarrow \text{edge}(\text{graph}) \cup \{C_3, C_4\} \)
14. for all cliques \((\alpha_1, \ldots, \alpha_{\mu-2})\) of size \( \mu - 2 \) such that for each \( \overline{\mu} \) with
\( \overline{\mu} \leq \mu \) the list of \( \overline{\mu} \) MNOLS generated by \((I, C_1, \alpha_1, \ldots, \alpha_{\overline{\mu}-2})\) is
set-canonical do
15. \( \mathcal{L} \leftarrow \) the list of \( \mu \) cyclic MNOLS of order \( n \) generated by
\((I, C_1, \alpha_1, \ldots, \alpha_{\mu-2})\)
16. \( \text{store} \leftarrow \text{store} \cup \{(\text{Is}_s(\mathcal{L}), \text{Is}_l(\mathcal{L}), \text{Red}_s(\mathcal{L}))\} \)
Table 2: The number of two MNOLS of order \( n \) under the given equivalence.

| \( n \) | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|
| set-isotopy | 1 | 2 | 9 | 68 | 1140 | 19040 | 489296 |
| set-reduced | 2 | 12 | 136 | 2340 | 52608 | 1589056 | 62516224 |
| list-isotopy | 1 | 3 | 12 | 128 | 2224 | 38000 | 977696 |
| list-reduced | 4 | 24 | 256 | 4640 | 105216 | 3178112 | 125026304 |

Table 3: The number of three MNOLS of order \( n \) under the given equivalence.

| \( n \) | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|
| set-isotopy | 0 | 1 | 1 | 73 | 4398 | 429111 | 70608753 |
| set-reduced | 0 | 6 | 16 | 2920 | 211104 | 36031716 | 9037728896 |
| list-isotopy | 0 | 2 | 6 | 438 | 26388 | 2574306 | 423652518 |
| list-reduced | 0 | 12 | 96 | 17520 | 1266624 | 216190296 | 54226373376 |

Table 4: The number of four MNOLS of order \( n \) under the given equivalence.

| \( n \) | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|
| set-isotopy | 0 | 0 | 0 | 0 | 1 | 2 | 117 | 14672 |
| set-reduced | 0 | 0 | 0 | 0 | 12 | 96 | 8638 | 1870592 |
| list-isotopy | 0 | 0 | 0 | 0 | 12 | 48 | 2484 | 350730 |
| list-reduced | 0 | 0 | 0 | 0 | 480 | 2304 | 207312 | 44879616 |

Table 5: The number of five MNOLS of order \( n \) under the given equivalence.

| \( n \) | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|
| set-isotopy | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

that are of type 0 increases as \( \mu \) increases. This may be of interest in future searches for sets of \( \mu \) MNOLS of order \( n \) where \( \mu \) is relatively large. Considering each type individually may allow more efficient construction of those set-isotopy classes with non-trivial set-autotopy group, as each set-autotopy must map row-intercalates to row-intercalates. Note that \( |\text{red}_s(\mathcal{L})| = 1 \) for \( n = 14 \), so we omit the column for \( |\text{red}_s(\mathcal{L})| \) in this case.
| $|\mathcal{I}_s(\mathcal{L})|$ | $|\mathcal{I}_l(\mathcal{L})|$ | #Type 0 | #Type 1 | #Total |
|---|---|---|---|---|
| 1 | 1 | 3618 | 15186 | 18804 |
| 2 | 1 | 0 | 80 | 80 |
| 2 | 2 | 46 | 88 | 134 |
| 3 | 3 | 2 | 14 | 16 |
| 6 | 6 | 1 | 5 | 6 |
| total: | | 3667 | 15373 | 19040 |

Table 6: The two MNOLS of order 14, by their type and autotopy group sizes.

| $|\mathcal{I}_s(\mathcal{L})|$ | $|\mathcal{I}_l(\mathcal{L})|$ | #Type 0 | #Type 1 | #Total |
|---|---|---|---|---|
| 1 | 1 | 202382 | 226436 | 428818 |
| 2 | 2 | 146 | 57 | 203 |
| 3 | 1 | 24 | 63 | 87 |
| 6 | 2 | 1 | 2 | 3 |
| total: | | 202553 | 226558 | 429111 |

Table 7: The three MNOLS of order 14, by their type and autotopy group sizes.

| $|\mathcal{I}_s(\mathcal{L})|$ | $|\mathcal{I}_l(\mathcal{L})|$ | #Type 0 | #Type 1 | #Total |
|---|---|---|---|---|
| 1 | 1 | 67 | 26 | 93 |
| 2 | 1 | 3 | 8 | 11 |
| 2 | 2 | 1 | 0 | 1 |
| 3 | 1 | 4 | 7 | 11 |
| 6 | 2 | 1 | 0 | 1 |
| total: | | 76 | 41 | 117 |

Table 8: The four MNOLS of order 14, by their type and autotopy group sizes.

| $|\mathcal{I}_s(\mathcal{L})|$ | $|\mathcal{I}_l(\mathcal{L})|$ | $|\text{red}_s(\mathcal{L})|$ | #Type 0 | #Type 1 | #Total |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 106794 | 380686 | 487480 |
| 2 | 1 | 1 | 12 | 822 | 834 |
| 2 | 2 | 1 | 260 | 660 | 920 |
| 4 | 2 | 1 | 0 | 12 | 12 |
| 2 | 1 | 2 | 46 | 0 | 46 |
| 4 | 2 | 2 | 4 | 0 | 4 |
| total: | | | 107116 | 382180 | 489296 |

Table 9: The two MNOLS of order 16, by their type and autotopy group sizes.
| $|S_s(L)|$ | $|S_l(L)|$ | $|red_s(L)|$ | #Type 0 | #Type 1 | #Total |
|-----|-----|-----|------|------|------|
| 1   | 1   | 1   | 36845488 | 33760273 | 70605761 |
| 2   | 2   | 1   | 2326 | 666 | 2992 |
| total: | 36847814 | 33760939 | 70608753 |

Table 10: The three MNOLS of order 16, by their type and autotopy group sizes.

| $|S_s(L)|$ | $|S_l(L)|$ | $|red_s(L)|$ | #Type 0 | #Type 1 | #Total |
|-----|-----|-----|------|------|------|
| 1   | 1   | 1   | 11146 | 3401 | 14547 |
| 2   | 2   | 1   | 79 | 79 | 107 |
| 2   | 1   | 2   | 8 | 8 | 8 |
| 4   | 1   | 4   | 1 | 0 | 1 |
| total: | 11190 | 3482 | 14672 |

Table 11: The four MNOLS of order 16, by their type and autotopy group sizes.

References

1. F. Demirkale, D.M. Donovan, J.L. Hall, A. Khodkar, and A. Rao. Difference covering arrays and pseudo-orthogonal latin squares. *Graphs and Combinatorics*, 2015. doi: 10.1007/s00373-015-1649-8.

2. F. Demirkale, D.M. Donovan, and A. Khodkar. Direct constructions for general families of cyclic mutually nearly orthogonal latin squares. *J. Combin. Des.*, 23(5):195203, 2015.

3. J. Egan and I.A. Wanless. Enumeration of mols of small order. *arXiv:1406.3681v2*, 2014.

4. I.A. Faradžev. Generation of nonisomorphic graphs with a given distribution of the degrees of vertices. (russian). *Algorithmic studies in combinatorics (Russian)*, 185:11–19, 1978.

5. A. Hulpke, P. Kaski, and P.R.J. Östergård. The number of latin squares of order 11. *Math. Comp.*, 80(274):1197–1219, 2011.

6. P. Kaski and P.R.J. Östergård. *Classification algorithms for codes and designs*. Classification Algorithms for Codes and Designs, Springer, 2006.

7. G. Kolesova, C.W.H. Lam, and L. Thiel. On the number of $8 	imes 8$ latin squares. *J. Combin. Theory Ser. A*, 54(1):143–148, 1990.

8. C. Laywine and G. Mullen. *Discrete mathematics using latin squares*. Wiley-Interscience Series in Discrete Mathematics and Optimization, 1998.
[9] P.C. Li and G.H.J van Rees. Nearly orthogonal latin squares. *J. Combin. Math. Combin. Comput.*, 62:1324, 2007.

[10] B.D. McKay. Isomorph-free exhaustive generation. *J. Algorithms*, 26(2):306–324, 1998.

[11] B.D. McKay and A. Piperno. Practical graph isomorphism, II. *J. Symbolic Computation*, 60:94–112, 2013. [http://dx.doi.org/10.1016/j.jsc.2013.09.003](http://dx.doi.org/10.1016/j.jsc.2013.09.003)

[12] B.D. McKay and I.M. Wanless. Most latin squares have many subsquares. *J. Combin. Theory Ser. A*, 86(2):322–347, 1999.

[13] B.D. McKay and I.M. Wanless. On the number of latin squares. *Ann. Comb.*, 9(3):335–344, 2005.

[14] B.D. McKay, A. Meynert, and W.J. Myrvold. Small latin squares, quasigroups and loops. *J. Combin. Des.*, 15(2):98119, 2007.

[15] E.B. Pasles and D. Raghavarao. Mutually nearly orthogonal latin squares of order 6. *Util. Math.*, 65:6572, 2004.

[16] D. Raghavarao, S.S. Shrikhande, and M.S. Shrikhande. Incidence matrices and inequalities for combinatorial designs. *Journal of Combinatorial Designs*, 10(1):1726, 2002.

[17] R.C. Read. Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976). Ann. Discrete Math.*, 2:107–120, 1978.

[18] G.H.J. van Rees. Private communication. October 2015.