THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Latin Squares and Related Structures

Trent Gregory Marbach

Bachelor of Science (Mathematics) - UQ

Bachelor of Science[Honours] (Mathematics) - UQ

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2015

School of Mathematics and Physics

Abstract

A *latin square* of order n is an $n \times n$ array of cells, each filled with one of n symbols such that each row and each column contain each symbol precisely once. This thesis contributes three new results from three different topics within the study of latin squares. In doing this, we give a broad overview of three distinct areas of the study of latin squares, and the results that have been accomplished in the literature so far.

The first study presents new results pertaining to transversals in latin squares. Previous work on transversals has investigated the spectrum of intersection sizes of two transversals within the back-circulant latin squares. A natural extension to this work is to investigate the spectrum of intersection sizes of more than two transversals within the back-circulant latin squares. In this thesis we will accomplish this for three and four transversals, and for all but a finite list of exceptions give a design theoretic construction that recursively builds from base designs that we found by a computational search.

The second study investigates μ-way k-homogeneous latin trades. These structures have been extensively studied when $\mu = 2$, but much less is known when $\mu > 2$. Previous investigation had filled in a fraction of the spectrum when $\mu = 3$. We continue this study giving new constructions and show that 3-way k-homogeneous latin trades of order m exist for all but 196 possible exceptions.

The third study investigates mutually nearly orthogonal latin squares (MNOLS). These MNOLS are similar to mutually orthogonal latin squares, and can also be used in the design of experiments. Continuing from previous investigations, we enumerate the number of collections of three cyclic MNOLS for latin squares with order up to 16. This required using computational enumeration techniques and a large optimised computer search, as

the search space was incredibly large. We present the number of collections of three MNOLS for latin squares with order up to 16 under a variety of equivalences, where we take the collections to be either sets or ordered lists.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Trent G. Marbach

Publications during candidature

These works were published during candidature:

1. T.G. Marbach, "On the intersection of three or four transversals of the back circulant latin square B_n", *Australasian Journal of Combinatorics* 65, No. 1, pp. 84-107, 2016.
2. T.G. Marbach, L. Ji, "The spectrum for 3-way k-homogeneous Latin trades", *Electronic Journal of Combinatorics* 22, Paper 4.1, 23pp, 2015.

Publications included in this thesis

These published works are incorporated in to the thesis:

1. T.G. Marbach, "On the intersection of three or four transversals of the back circulant latin square B_n", *Australasian Journal of Combinatorics* 65, No. 1, pp. 84-107, 2016. Incorporated as Chapter 2.

   Contributions: Marbach 100% of the key ideas, 100% of the technical calculations, 100% of the drafting and writing, 100% of initial editing.

2. T.G. Marbach, L. Ji, "The spectrum for 3-way k-homogeneous Latin trades", *Electronic Journal of Combinatorics* 22, Paper 4.1, 23pp, 2015. Incorporated as Chapter 3.

   Contributions: Marbach approximately 50% of the key ideas, 100% of the technical calculations, 100% of the drafting and writing, approximately 50% of initial editing. Ji approximately 50% of the key ideas, approximately 50% of initial editing.

Contributions by others to the thesis

My supervisors Diane M. Donovan and Benjamin A. Burton helped in guiding me towards suitable problems and in technical editing. Ji Lijun and Zhu Lie worked with me on the construction of $(3, k, m)$-latin trades via RPBD's. We would like to acknowledge Ian Wanless for comments that helped us to improved section 2 of Chapter 2. Diane M. Donovan and Fatih Dermikale worked with me on Chapter 4.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgements

Firstly I would like to show my deepest appreciation for my principal supervisor Diane Donovan. Her guidance, patience, encouragement, and kindness has always been amazing. Beyond leading me through my work as a PhD candidate, she has more than anybody else taught me how to think, for which I am deeply indebted to her.

To my associated supervisor Benjamin Burton, who encouraged me and gave me vital direction. To the Mathematics Discipline at the University of Queensland, whose members have been constantly friendly and wise. I would also like to acknowledge the financial assistance provided by the Commonwealth Government by way of an Australian Postgraduate Award.

For an 8 month period I was hosted at Soochow University, Suzhou, China. I would like to acknowledge the large amounts of support and guidance that was given to me from Ji Lijun during that time. It was an amazing experience being able to research at such a beautiful campus with a number of great mathematicians. Also to Zhu Lie for his mathematical guidance during this time.

To my family, who have always supported me in every endeavor I have embarked upon. Particularly my father for being my rock, Cora for her continual guidance, Tanya for being the best role model, Tristan for always showing me the fun side of life, and Monique for her companionship.

Keywords

latin square, latin trade, transversal, nearly orthogonal, k-homogeneous, intersection problem, combinatorial enumeration.

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 010104, Combinatorics and Discrete Mathematics (excl. Physical Combinatorics), 100%

Fields of Research Classification

FoR code: 0101, Pure Mathematics, 100%

Contents

List of Figures

List of Tables

Index

Chapter 1

Introduction

1.1 Brief History

A *latin square* of order n is an $n \times n$ array of symbols such that each symbol occurs once in each row and once in each column. Examples of latin squares appeared as early as the 12th century. The writer Ahmad ibn 'Ali ibn Tusuf al-Buni (d. 1225) commented on their use on talismans, as well as giving a hint towards a possible construction for magic squares (magic squares significantly predated this usage of latin squares, having been known to the Islamic world since the 9th Century and in China since the 4th century BCE, see [33] page 525). They also appeared in the work of the 13th century Spanish mystic and philosopher Ramon Lull, in a 1356 book of Indian mathematics, and in a book of recreational card problems in the early 18th century (See [6] for further details of this history, and references).

The systematic study of latin squares began in a paper by Leonhard Euler. This paper [44] began with the now famous thirty-six officers problem (translation via [43]):

> "Une question fort curieuse qui a exercé pendant quelque temps la sagacité de bien du monde, m'a engagé à faire les recherches suivantes, qui semblent ouvrir une nouvelle carrière dans l'Analyse et en particulier dans la doctrine des combinaisons. Cette question rouloit sur une assemblée de 36 Officiers de six différens grades et tirès de six Régimens différens, qu'il s'agissoit de ranger dans un quarré de manière que sur chaque ligne, tant horizontale que verticale, il se trouvât six Officiers tant de différens caractères que de Régimens différens."

> "A very curious question that has taxed the brains of many (has) inspired me to undertake the following research that has seemed to open a new path in Analysis and in particular in the area of combinatorics. This question concerns a group of thirty-six officers of six different ranks, taken from six different regiments, and arranged in a square in a way such that in each row and column there are six officers, each of a different rank and regiment."

Euler asked the more general question of whether there exist a pair of latin squares L_1 and L_2 of order n with the property that the superimposition of L_1 onto L_2 yields all n^2 possible pairs of symbols amongst the cells of the superimposition. When such a property holds, L_1 and L_2 are said to be *orthogonal*. For a latin square L, let the symbol appearing in row r and column c be denoted $L(r, c)$. Then the thirty-six officers problem, if possible, would correspond to a pair of orthogonal latin squares L_1 and L_2 of order 6, where the officer in row r and column c has rank $L_1(r, c)$ and regiment $L_2(r, c)$. It turns out that such a configuration is impossible. Euler conjectured that there did not exist a pair of orthogonal latin squares when $n \equiv 2 \pmod 4$. This was later shown to be false [13] [12], and in fact it turns out that examples of pairs of orthogonal latin squares exist for all n, except for $n = 2, 6$ [14].

This problem is naturally generalized to asking for s mutually orthogonal latin squares (MOLS) of order n. Although two MOLS of order 10 have been found, it is still unknown whether three MOLS of order 10 exist. Similarly, four MOLS of order 14 have been found in [81], but it is still unknown whether five MOLS of order 14 exist.

The study of latin squares has also grown independently of the study of mutually orthogonal latin squares. Interest in latin squares has also been spurred by the connections found to algebra (through quasigroups, a generalisation of a group) and to statistics (through the design of experiments) in the 1930's. In this thesis we will explore a number of current questions related to latin squares, and provide new results in these areas. We will fully explain each of the subjects dealt with in this thesis, but first we are required to make an interlude to formalize our notation.

1.2 Definitions

A *partial latin square of order* n, $T = [t(r,c)]$, is an $n \times n$ array of cells with each cell either filled with an element $t(r,c)$ of Ω (a set of n symbols) or left empty, such that each symbol of Ω appears at most once in each row, and at most once in each column. In what follows, we typically take $\Omega = [0, n-1] = \{0, 1, 2, \ldots, n-1\}$, and index the rows and columns of the partial latin square by Ω. A major exception to this is in Chapter 3, where we take $\Omega = [n] = \{1, \ldots, n\}$. A partial latin square T has *volume* s if it has precisely s filled cells, where $0 \leq s \leq n^2$. A partial latin square with volume n^2 is a *latin square*. We can represent T as a set of s ordered triples $\{(r, c, t(r,c)) \mid r, c \in \Omega \text{ and cell } (r,c) \text{ is not empty}\}$.

A commonly studied latin square is the back circulant latin square, which is defined as $B_n = \{(r, c, r+c \mod n) \mid r, c \in [0, n-1]\}$. The latin squares B_n have a strong connection to diagonally cyclic latin squares, and are often used to prove facts about latin squares in general.

A *diagonal* of a latin square L is a set of n cells of L such that each row and each column is represented in the set of cells. A *transversal* of a latin square is a diagonal that also has each symbol of Ω represented in the diagonal.

Consider a partial latin square $T \subset \Omega \times \Omega \times \Omega$. The *shape* of T is defined as $\mathcal{S}(T) = \{(r,c) \in \Omega \times \Omega \mid (r,c,e) \in T, \text{ for some } e \in \Omega\}$. The *rth row set* of T is defined as $\mathcal{R}_r(T) = \{e \in \Omega \mid (r,c,e) \in T, \text{ for some } c \in \Omega\}$. The *cth column set* of T is defined as $\mathcal{C}_c(T) = \{e \in \Omega \mid (r,c,e) \in T, \text{ for some } r \in \Omega\}$. The *eth symbol set* of T is defined as $\mathcal{E}_e(T) = \{(r,c) \in \Omega \times \Omega \mid (r,c,e) \in T\}$.

Definition 1.2.1. *For natural numbers μ, n, $\mu \le n$, a μ-*way latin trade of order n on symbol set Ω *is a collection $\mathcal{T} = (T_1, \dots, T_\mu)$ of μ partial latin squares of order n using symbols of Ω such that:*

- $\mathcal{S}(T_\alpha) = \mathcal{S}(T_\beta)$, *for each* $1 \le \alpha < \beta \le \mu$;
- *for each* $(r,c) \in \mathcal{S}(T_\alpha)$ *it holds that* $t_\alpha(r,c) \ne t_\beta(r,c)$, *for every* $1 \le \alpha < \beta \le \mu$; *and*
- $\mathcal{R}_r(T_\alpha) = \mathcal{R}_r(T_\beta)$ *and* $\mathcal{C}_c(T_\alpha) = \mathcal{C}_c(T_\beta)$, *for each* $r, c \in \Omega$ *and* $1 \le \alpha < \beta \le \mu$.

In the case that $\mu = 2$, the term *latin bitrade* is often used, however we will make rare use of it.

Let $\mathcal{T} = (T_1, \dots, T_\mu)$ be a μ-way latin trade. As the shape of each T_α is the same, we can define the shape of $\mathcal{T}$ as $\mathcal{S}(\mathcal{T}) = \mathcal{S}(T_1)$. Then the volume of $\mathcal{T}$ is the volume of T_1. Similarly, the row sets (resp. column sets) of each T_α are the same, so we can define a row set for row r (resp. column set for column c) of $\mathcal{T}$ as $\mathcal{R}_r(\mathcal{T}) = \mathcal{R}_r(T_1)$ (resp. $\mathcal{C}_c(\mathcal{T}) = \mathcal{C}_c(T_1)$).

Definition 1.2.2. *For an integer $k \geq 0$, a (μ, k, m)-*latin trade *on symbol set Ω, $\mathcal{T} = (T_1, \dots, T_\mu)$, is a μ-way latin trade of order m on Ω that has $k = |\mathcal{R}_r(\mathcal{T})| = |\mathcal{C}_c(\mathcal{T})| = |\mathcal{E}_e(T_1)|$, for each $r, c, e \in \Omega$. Such a μ-way latin trade is called k-*homogeneous.

Note that $|\mathcal{E}_e(T_1)| = |\mathcal{E}_e(T_\alpha)|$ for $2 \leq \alpha \leq \mu$ as $\mathcal{T}$ is a μ-way latin trade, for each $e \in \Omega$. So each symbol appears k times in each partial latin square of a (μ, k, m)-latin trade. A (μ, k, m)-latin trade can have $k = 0$ in the case that each of the μ partial latin squares is empty; otherwise k must satisfy $\mu \leq k \leq m$.

A *latin trade* T of a latin square L is a partial latin square such that $T \subseteq L$ and T is one of the two partial latin squares in a 2-way latin trade. If the 2-way latin trade that T appears in is k-homogeneous, we say T is k-homogeneous. If T contains no other latin trade as a proper subset, it is *minimal*. Given T' with (T, T') a 2-way latin trade, we call T' a *disjoint mate* of T.

A pair of latin squares L_1, L_2 of order n are called *orthogonal* if $\{(L_1(r, c), L_2(r, c)) \mid r, c \in \Omega\} = \Omega \times \Omega$. A set of μ latin squares are *mutually orthogonal* if they are pairwise orthogonal, and we refer to such a set as a set of *MOLS*.

A pair of latin squares L_1, L_2 of even order n are called *nearly orthogonal* if it holds that $\{(L_1(r, c), L_2(r, c)) \mid r, c \in \Omega\} = \Omega \times \Omega \setminus \{(e, e) \mid e \in \Omega\}$ and for each $e \in \Omega$ there are pairs $(r_1, c_1) \neq (r_2, c_2)$ such that $(e, e + n/2 \mod n) = (L_1(r_1, c_1), L_2(r_1, c_1)) = (L_1(r_2, c_2), L_2(r_2, c_2))$. Less formally, this says 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 will deal with both sets and lists of pairwise nearly orthogonal latin squares.

1.3 Questions

1.3.1 Transversals in latin squares

A recent computational search seeking three MOLS of order 10 [69] focused on a small subset of the possible latin squares (a latin square for each of the $8,500,842,802$ main classes of latin squares of order 10 with a non-trivial autoparatopy group. See the paper for definitions and details). This restriction was necessary due to the enormous number of latin squares of order 10 (see [33]). The search did not yield a set of three MOLS of order 10, but it was the largest search to date. Even having placed this heavy restriction on the latin squares involved, this calculation took 172 years of CPU time. A fact in this search was the well known result that a latin square L has an orthogonal mate if and only if L is decomposable into transversals (see [33]). This search constructed a graph based on transversals on each the aforementioned latin squares, and looked for specific cliques within this graph.

Such an approach in the study of MOLS is not uncommon. The number of transversals in a latin square has been used as a heuristic to identify latin squares that may have a large number of orthogonal mates (see [15], although this is just a heuristic, as some latin squares with a large number of transversals do not have an othogonal mate [68], and some latin squares with a small number of transversals do have an orthogonal mate [83]). Another result about transervals showed that for $n > 3$, there exists a latin square of order n with a cell that is contained in no transversal of the latin square [87]. This implies that there is always some latin square of order $n > 3$ without an orthogonal mate.

The study of transversals has been encouraged by this connection to MOLS, although the study of transversals has yielded many stand-alone results. Further work on transversals has investigated (to name a small number of topics) the largest size of a partial transversal in any latin square of a given order (see [85] for details), the minimum and maximum number of transversals amongst all the latin squares of a given order (again, see [85]), and the possible intersection size of two transversals in a latin squares (in particular, using transversals in the back-circulant latin square B_n [30]).

The back-circulant latin square B_n has many interesting connections to other structures. Most relevant to this thesis is that a transversal of B_n is equivalent to a diagonally cyclic latin square of order n [84]. A transversal of B_n is also equivalent to a complete mapping of the cyclic group of order n as well as an orthomorphism of the cyclic group of order n [28] (see also [85]). (Other equivalences can be found in [30].) Properties of B_n have also been used to help identify properties about latin squares in general. For example, the smallest known critical set in a latin square of order n (see next section for a definition of a critical set) is a critical set in B_n [37]. Also, the largest known number of transversals in a latin square are those in B_n [30].

Given that the problem of finding the possible intersection size of two transversals in B_n is of interest, we ask the following question:

Question 1.3.1. *For what t does there exist a collection of μ transversals of the back circulant latin square of order n, B_n, such that each pair of transversals intersect precisely in the same t cells?*

We will answer this question in Chapter 2 for all but a finite list of exceptions in the cases of $\mu = 3$ and $\mu = 4$. The results of this work are documented in [63].

1.3.2 μ-way k-homogeneous latin trades

Interest in latin trades arose from the study of the intersection problem for latin squares and of critical sets for latin squares.

The study of the intersection problem for latin squares began in [51]. The initial problem asked for the spectrum of sizes of the intersection of two latin squares. A modification to this problem is finding the spectrum of sizes of the intersection of more than two latin squares, which is called the μ-way intersection problem for latin squares. In this case, we say a set of three or more latin squares have common intersection S if any pair of the latin squares from the set of latin squares has intersection S. The spectrum of possible common intersection sizes $|S|$ has been investigated for sets of three and four latin squares (for three latin squares with common intersection, see [1] [46]; for four latin squares with common intersection, see [4]). Another modification to the initial problem that has found some attention is asking for the spectrum of possible sizes of the intersection of two or more latin squares with certain properties (i.e. totally symmetric latin squares [49]; semi-symmetric latin squares [48]; commutative latin squares [47]; idempotent latin squares, idempotent commutative latin squares, and latin cubes [50]; latin squares of different orders [41]; and latin squares whose difference is a collection of m-flowers [60] [66]).

To see the connection between the intersection problem for latin squares and latin trades, notice that if the intersection of two latin squares L_1 and L_2 is the set of cells S, then removing S from the latin squares yields the 2-way latin trade $\{L_1 \setminus S, L_2 \setminus S\}$. Although originally only the intersection of latin squares was studied, a shift in thinking recognized the study of latin trades in its own right. Studying the possible volumes of μ-way latin trades is then a problem with strong connections to the μ-way intersection problem for latin squares, although these problems are not identical. As the Cayley table of a quasigroup is equivalent to a latin square, latin trades first appeared as the equivalent exchangeable partial groupoid [40], and have also appeared by other names (i.e. latin interchanges, critical partial latin squares).

A *critical set* P of a latin square L is a partial latin square that is a subset of precisely one latin square, namely L, and is minimal in this property. That is, if we remove any filled cell from P, then the resulting partial latin square is a subset of two or more latin squares. Amongst the set of all latin squares of order n, it is of interest to find the smallest size of a critical set, which is denoted $\mathrm{scs}(n)$, and also the largest size of a critical set, which is denoted $\mathrm{lcs}(n)$.

Latin trades have played a large role in the search for critical sets. This is due to the fact that any critical set of L must intersect every latin trade of L (see [33]). As any two rows of a latin square form a latin trade, then any critical set of the latin square must intersect at least one of those two rows. This gives a simple lower bound of $\mathrm{scs}(n) \geq n-1$. It was suggested [9] that studying sets of 3 or 4 rows of a latin square may yield an improvement on the bound, and subsequently [22] showed that in any set of three rows of any latin square there exists a latin trade with an empty column, and used this to provide a new bound of $\mathrm{scs}(n) \geq 2n-4$. More complicated trades have been investigated [19], and using these trades the bound was improved to $\mathrm{scs}(n) \geq n\lfloor(\log n)^{1/3}/2\rfloor$, which is asymptotically better. An upper bound on $\mathrm{scs}(n)$ has also been found [34][79] giving $\mathrm{scs}(n) \leq \lfloor n^2/4 \rfloor$, which was achieved by finding an example of a critical set in B_n for each n. It is a conjecture [33] that in fact $\mathrm{scs}(n) = \lfloor n^2/4 \rfloor$ for all n. There has also been some investigation of the spectrum of possible sizes of critical sets [26][38] using latin trades to verify that the set of cells is a critical set. Lower and upper bounds on $\mathrm{lcs}(n)$ have been found by both constructive and non-constructive techniques, although we will not endeavor to detail them here (for a survey on critical sets in latin squares see [56], and for details of some more recent work see [23]).

The latin trades that we study in this thesis have the k-homogeneous property. One of the first papers to study the k-homogeneous property for latin trades was [24], which in particular studied $(2,3,n)$-latin trades. These $(2,3,n)$-latin trades were constructed by hexagonal packings of the plane with circles. It was later shown that the $(2,3,n)$-latin trades constructed in this manner classify every minimal $(2,3,n)$-latin trade, and that any $(2,3,n)$-latin trade can be decomposed into three disjoint partial transversals [20]. (See also [39] for a generalization of these kinds of trades, called homogeneous toroidal latin bitrades.) The fact that $(2,k,n)$-latin trades can be decomposed into disjoint partial transversals is interesting (see [27] for further work on this topic, and [29] for an investigation of partial transversals of k-plexes, a similar structure to $(2,k,n)$-latin trades). It was found that $(2,k,m)$-latin trades can be embedded into abelian 2-groups [21]. It was also pointed out that latin trades intersecting a large number of rows and columns while having a relatively small volume are useful, and k-homogeneous trades could be said to have this property, especially when k is relatively small to n.

As we mentioned earlier, a transversal of B_n is equivalent to a diagonally cyclic latin square of order n. Using this fact, in Chapter 2 we use transversals to construct certain $(3,k,m)$-latin trades. This leads us into the work of Chapter 3, where we investigate the question posed by [7]:

Question 1.3.2. *For given m and k, $\mu \leq k \leq m$, does there exist a (μ,k,m)-latin trade?*

We will use a number of constructions along with previous results to answer Question 1.3.2 in the case that $\mu = 3$ for all but a finite list of exceptions. The results of this work have been documented in [64].

1.3.3 Mutually nearly orthogonal latin squares

Mutually orthogonal latin squares have been used (under the name row-column designs) to design experiments to investigate the covariance between experimental factors. For instance an experimenter might wish to study two medications, say drug A and drug B, where the two medications have some potential cross-over effects. Suppose that the drugs can be administered at one of n different dosage levels, that there are n subjects taking the medication, and a trial lasts for n dosage days. Take $\{L_1, L_2\}$ to be a pair of MOLS of order n. Suppose subject s on day d takes dosage level $L_1(s,d)$ of drug A and dosage level $L_2(s,d)$ of drug B. Then over the trial, each subject will have taken each possible dosage of each drug precisely once. Further, each possible pairing of dosage levels has occurred precisely once amongst the subjects.

Raghavarao, Shrikhande, and Shrikhande [76] introduced the concept of mutually nearly orthogonal latin squares (MNOLS), modifying the typical orthogonal property of latin squares. These MNOLS can be used in place of MOLS in the design of experiments, and their introduction was based on two possible benefits: First, as in the case of drug dosages, there may be cases when subjects should not receive interventions that are all at the maximum level or all at the minimum level. As the pairs of difference zero do not appear in MNOLS, such cases are avoided when MOLS are replaced by MNOLS. Second, there are some cases when conducting an experiment requiring sets of MOLS of certain sizes that are hard or impossible to find, and so it may be necessary to find a set of latin squares that are 'close' to a set of MOLS. As we mentioned before, it is known that two MOLS of order 6 do not exist, and it is unknown whether three MOLS of order 10 exist. However three MNOLS of order 6 exist [76] as do three [74] and even four MNOLS of order 10 [61]. While the maximum size of a set of MOLS of order n when n is a prime power is well known (see for example [33]), asking for the maximum size of such a set when n is not a prime power is usually difficult. It appears that MNOLS may be a good replacement for MOLS in such cases.

In this thesis, we concern ourselves with both the existence and enumeration of MNOLS. We will consider both ordered lists and sets of μ MNOLS of order n, recalling that n must be even for the definition of nearly orthogonal to make sense. Note that the distinction between lists and sets of MNOLS is only important during enumeration, and not while investigating existence. Finding the existence and enumeration of MNOLS has had some prior attention. A simple constuction found that two MNOLS of order n exist for all n

even [74]. It has been found that there also exists three MNOLS of order n for each even $n \geq 6$, except perhaps when $n = 146$ (although the effort required for this was much greater, see [36][35][61][76]). It was also shown [75] that there does not exist four MNOLS of order 6. It is interesting to note that difference covering arrays can be used to construct μ MNOLS as in [36].

A list (set) of μ MNOLS of order n is *cyclic* if each latin square L in the list (set) has $(r, c, e) \in L$ if and only if $(r, c+1 \pmod{n}, e+1 \pmod{n}) \in L$ for all $r, c \in [0, n-1]$. An approach to finding sets of μ MNOLS that has been successful is to construct μ cyclic MNOLS. Throughout the literature there has been some attempt to enumerate the μ cyclic MNOLS under certain equivalences. The most recent paper to investigate this enumeration [61] found the number of non-equivalent sets up to isotopism (see Chapter 4 for the definition of isotopism) of μ cyclic MNOLS of order n for $n \leq 12$. The number of these sets of μ cyclic MNOLS of order n is given in table 1.1.

n	6	8	10	12
$\mu = 3$	1	1	≥ 1	> 1
$\mu = 4$	0	0	1	> 1
$\mu = 5$	0	0	0	0

Table 1.1: The number of sets of μ MNOLS of order n under isotopic equivalence.

Given the rather small number of inequivalent MNOLS that are known to exist, and the incompleteness of some of the counts, it will be useful to extend the previous results, and in this thesis we do this to $n \leq 16$. There are two questions of significance that we focus on:

Question 1.3.3. *For a fixed n, what is the largest μ such that a set of μ* MNOLS *of order n exists?*

Question 1.3.4. *For a fixed μ and n, how many distinct collections of μ* MNOLS *of order n exists?*

In Chapter 4, we answer Question 1.3.4 for $2 \leq \mu \leq 5$ and $n \leq 16$ under a variety of equivalences. This will also in turn answer Question 1.3.3 for $n \leq 16$, and will resolve in the negative a conjecture of [61] that proposed the maximum μ for which a set of μ cyclic MNOLS of order n exists is $\lceil n/4 \rceil + 1$. The results of this work have been documented in [65].

Chapter 2

Transversals in the back circulant latin squares

2.1 Introduction

A natural question to ask in combinatorics is how may two distinct examples of a certain combinatorial structure intersect, a question which has been investigated for a large variety of different structures. An extension of this is to consider the μ-way intersections of the structures, and work has been done taking the underlying structure to be Steiner Triple Systems in [73], m-cycle systems in [2], and latin squares in [3] and [1].

There has been an investigation into the possible intersection size of two transversals of the back circulant latin square [30], and so in a similar fashion we generalize from the intersection of two transversals to the intersection of a collection of μ transversals. See [86] for a survey of transversals in latin squares.

The problem that this chapter investigates is Question 1.3.1, which we restate for convenience:

Question. *For what t does there exist a collection of μ transversals of the back circulant latin square of order n, such that each pair of transversals intersect precisely in the same t points?*

These transversals can be used to construct (μ, k, n)-latin trades with n odd, which we will use later in Chapter 3. We will sometimes reference rows, columns and symbols with indices that are greater than $n-1$, by which we will always mean the representation of this index modulo n.

Throughout this chapter, we assume n is odd, as it is well known that B_n contains no transversals for any even n. The possible intersection sizes of any two transversals of B_n has been determined:

Theorem 2.1.1. [30] *For each odd n, there exists a pair of transversals of B_n that intersect in t cells, when $n \neq 5$ for $t \in \{0, \ldots, n-3\} \cup \{n\}$, and when $n = 5$ for $t \in \{0, 1, 5\}$.*

We consider a generalization of such intersections of pairs of transversals to the intersection of μ transversals.

Definition 2.1.2. *A collection of μ transversals $T_1, \ldots, T_\mu$ intersect* stably *in t points if there is a set $S \subseteq [0, n-1]^2$ of size $|S| = t$ such that $S = \cap_{i=1}^{\mu} T_i$ and $\emptyset = (T_i \cap T_j) \setminus S$ for each $1 \leq i < j \leq \mu$.*

Informally, if there is a cell $(i, j, k) \in S$, then (i, j, k) appears in each transversal $T_1, \ldots, T_\mu$. If there is a cell $(i', j', k') \in T_\alpha$ with $(i', j', k') \notin S$, then no other transversal contain (i', j', k').

Then Question 1.3.1 is asking for what values of t does there exist a collection of μ transversals of B_n that intersect stably in t points. The main results of this chapter are the following two theorems:

Theorem 2.1.3. *For odd $n \geq 33$, let I, I', d and d' be the unique integers such that $n = 18I + 9 + 2d$ and $n = 22I' + 11 + 2d'$, $I, I' \geq 1$, $0 \leq d < 9$ and $0 \leq d' < 11$. Then there exist three transversals of B_n that intersect stably in t points for $t \in [\min(3 + d', d), n] \setminus [n-5, n-1]$ except, perhaps, when:*

- $n = 51$ *and* $t = 29$,
- $n = 53$ *and* $t = 30$.

Theorem 2.1.4. *For odd $n \geq 33$, let I, I', d and d' be the unique integers such that $n = 18I + 9 + 2d$ and $n = 22I' + 11 + 2d'$, $I, I' \geq 1$, $0 \leq d < 9$ and $0 \leq d' < 11$. Then there exist four transversals of B_n that intersect stably in t points for $t \in [\min(3 + d', d), n] \setminus (\{n-15\} \cup [n-7, n-1])$ except, perhaps, when:*

- $33 \leq n \leq 43$ *and* $t \in [10 + d', 11 + d'] \cup [n-14, n-12]$,
- $45 \leq n \leq 53$ *and* $t \in [-1 + d', 2 + d'] \cup [10 + d', 11 + d'] \cup [18 + d', 20 + d']$,
- $63 \leq n \leq 75$ *and* $t \in [7 + d, 8 + d]$.

2.2 Results

2.2.1 Basic results

Lemma 2.2.1. *For an odd integer n, there exists a collection of μ transversals of B_n which intersect stably in n points, for any $\mu \geq 1$.*

Proof. For odd n, the main diagonal's cells $(i, i, 2i)$ with $i \in [0, n-1]$ form a transversal of B_n, showing at least one transversal exists. A collection of μ identical transversals intersects stably in n points. $\square$

Lemma 2.2.2. *For an odd integer n, there exists a collection of μ transversals of B_n which intersect stably in 0 points, for any $1 \leq \mu \leq n$.*

Proof. Consider the μ transversals of B_n given by $T_\alpha = \{(i, i+\alpha, 2i+\alpha \mid i \in [0, n-1]\}$ for $\alpha \in [0, \mu - 1]$. These μ transversals intersect stably in 0 points. □

Lemma 2.2.3. *For odd μ, there exists a collection of μ transversals of B_n which intersect stably in $n - \mu$ points if and only if $\mu \mid n$.*

Proof. Suppose there exists a collection of μ transversals of B_n which intersect stably in $n - \mu$ points. Define R (resp. C) as the set of rows (columns) that have no pair of transversals intersecting in those rows (columns). Then the cells of the set $H = \{(r, c, e') \in T_\alpha \mid 1 \leq \alpha \leq \mu, r \in R \text{ and } c \in C\}$ can only be filled with one of μ distinct symbols, as the other $n - \mu$ symbols appear in the stable intersection of the transversals. So H forms a $\mu \times \mu$ subsquare of B_n. Theorem 3 of [17] tells us such a subsquare implies the cyclic group of order n has a subgroup of order μ. Then μ must divide n.

Now suppose $\mu \mid n$, so $n = m \cdot \mu$ for some integer m. Let $T^1_\alpha = \{(mi, m(i+\alpha), m(2i+\alpha)) \mid 0 \leq i \leq \mu - 1\}$ and $T^2 = \{i, i, 2i \mid 0 < i \leq n-1 \text{ and } m \nmid i\}$, for $\alpha \in [1, \mu]$. Define transversals $T_\alpha = T^1_\alpha \cup T^2$ for $\alpha \in [1, \mu]$. Then $T_\alpha \cap T_\beta = T^2$, for each $1 \leq \alpha < \beta \leq \mu$, where $|T^2| = n - \mu$. □

Lemma 2.2.4. *For odd integers n, m, if $m \mid n$ and for integers q with $0 \leq q \leq n/m - 1$ there exists μ transversals of B_m that intersect stably in t_q points, then there exists μ transversals of B_n that intersect stably in $\sum_{q=0}^{n/m-1} t_q$ points.*

Proof. We construct μ transversals of B_n by combining μ transversals chosen from each of the subsquares $S_q = \{(mi + q, m(i+\alpha) + q, m(2i+\alpha) + 2q) \mid 0 \leq i \leq \mu - 1\}$ for $0 \leq q \leq n/m - 1$. Each of these are subsquares of B_n that are equivalent to B_m, and so for each S_q we use the μ transversals of B_m that intersect stably in t_q points. Combining these n/m collections of μ transversals of size m gives μ transversals of B_n that intersect stably in $\sum_{q=0}^{n/m-1} t_q$ points. $\square$

Lemma 2.2.5. *For an odd integer n, there does not exists a collection of μ transversals of B_n that intersect stably in t points, for $t \in \{n-\mu+1, \ldots, n-1\}$, for any $\mu \geq 2$.*

Proof. Suppose that there exists a collection of μ transversals that intersect stably in t points, for $t \geq 1$. Let $C \subseteq [0, n-1]$ be the set of columns such that no pair of transversals of our collection of μ transversal intersect in column $c \in C$. If row r has no pair of transversals intersecting in row r, then the set $\{(r, c', e') \in T_\alpha \mid 1 \leq \alpha \leq \mu \text{ and } c' \in C\}$ has size μ. But this implies $|C| \geq \mu$, which means there can be at most $n - \mu$ columns where the μ transversals meet. This implies the result. $\square$

2.2.2 Computer search

We performed a computer search for μ transversals of B_n when n is relatively small, and $\mu = 3, 4$. For $n \in \{5, 7, 9, 11, 13\}$, the program was able to exhaustively check the search space for both $\mu = 3, 4$, and also $n = 15$ for $\mu = 3$. For the other odd $n \leq 31$, we were only able to obtain partial results, as the search space was quite large. The results are summarized in Tables 2.1 and 2.2.

2.2.3 Principal construction

For this section, take n to be a fixed odd integer. Define $B^n_{i,j}$ to be the $j \times j$ subsquare of B_n at the intersection of rows and columns with indices $i, i+1, \ldots, i+j-1$. We will write $B_{i,j}$ instead of $B^n_{i,j}$ when the value of n is clear in the given context. The cells of such a subsquare are filled with symbols from $\{2i, \ldots, 2i+2j-2\}$. We consider a partial transversal within the $j \times j$ subsquare of cells $B_{i,j}$ to be a set of j triples

n	t
5	1
7	1,2
9	1,2,3,4,6
11	1,2,3,4,5,6
13	1,2,3,4,5,6,7
15	1,2,3,4,5,6,7,8,9,10,11,12
17	1,2,3,4,5,6,7,8,9,10,11
19	1,2,3,4,5,6,7,8,9,10,11,12,13
21	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,18
23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
25	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,21
27	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22
31	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24

Table 2.1: There exists a collection of $\mu = 3$ transversals of B_n with stable intersection size t.

n	t
7	1
9	1,2
11	1,2,3
13	1,2,3,4
15	1,2,3,4,5,10
17	1,2,3,4,5,6
19	1,2,3,4,5,6,7
21	1,2,3,4,5,6,7,8
23	1,2,3,4,5,6,7,8,9
25	1,2,3,4,5,6,7,8,9,10
27	1,2,3,4,5,6,7,8,9,10,11
29	1,2,3,4,5,6,7,8,9,10,11,12
31	1,2,3,4,5,6,7,9,10,11,12,13

Table 2.2: There exists a collection of $\mu = 4$ transversals of B_n with stable intersection size t.

$\{(r_k, c_k, e_k) \mid r_k - i, c_k - i \in [0, j-1] \text{ and } 0 \le k \le j-1\}$ such that $|\{r_k \mid 0 \le k \le j-1\}| = |\{c_k \mid 0 \le k \le j-1\}| = |\{e_k \mid 0 \le k \le j-1\}| = j$. We further consider a set of subsquares $B_{i,j}$ such that the subsquares partition the rows and columns of B_n. Then finding certain collections of μ partial transversals within each of these $B_{i,j}$ that only use certain symbols will amount to finding a collection of μ transversals of B_n.

Fix μ an integer and b an odd integer with $3 \le b \le n/3$. For a given $\bar{d}$ and t, we write $t \in \Omega_\mu^b(b+\bar{d})$ if there exists a collection of μ partial transversals of $B_{0,b+\bar{d}}$ within B_n that intersect stably in t points, and only use symbols from $\{(b-1)/2, \dots, 3(b-1)/2 + 2\bar{d}\} \setminus \{b + 2j \mid 0 \le j < \bar{d}\}$. Notice that $B_{i,j}$ is just a relabeling of the symbols of $B_{0,j}$, and so the existence of a collection of μ partial transversals within $B_{0,j}$ is equivalent to the existence of a collection of μ partial transversals within $B_{i,j}$.

Take I and d to be the unique integers with $I \ge 1$, $d \in [0, b-1]$, and $n = 2Ib + b + 2d$. We consider three types of subsquares; large subsquares $B_{0,b+d}$, small subsquares $B_{(n+b)/2,d}$ and base subsquares $B_{ib+d,b}$ and $B_{(I+i)b+2d,b}$ for $1 \le i \le I$. Figure 2.1 shows the layout of the subsquares. The symbols that fill the cells of the partial transversal from each of these subsquares are restricted. In particular, the base subsquares $B_{ib+d,b}$ use the symbols $\{2(ib+d)+(b-1)/2, \dots, 2(ib+d)+3(b-1)/2\}$ and the base subsquares $B_{(I+i)b+2d,b}$ use the symbols $\{2(Ib+ib+2d)+(b-1)/2, \dots, 2(Ib+ib+2d)+3(b-1)/2\}$ for $1 \le i \le I$, the large subsquare $B_{0,b+d}$ use the symbols $\{(b-1)/2, \dots, 3(b-1)/2 + 2d\} \setminus \{b + 2j \mid 0 \le j < d\}$, and the small subsquare $B_{(n+b)/2,d}$ use the symbols $\{b + 2j \mid 0 \le j < d\}$.

These symbols have been chosen so that the partial transversal of one of the subsquares does not share any symbols in common with the partial transversal of any other subsquare. We demonstrate the interleaving that occurs for the base subsquares in Figure 2.2.

Theorem 2.2.6. *Let n, b be odd integers, $3 \le b \le n/3$, and μ an integer with $\mu \ge 2$. Let $I \ge 1$ and $d \in [0, b-1]$ be the unique integers such that $n = 2bI + b + 2d$. There exists μ transversals of B_n that intersect stably in t points with $t = d + \sum_{i=0}^{2I} t_i$, where*

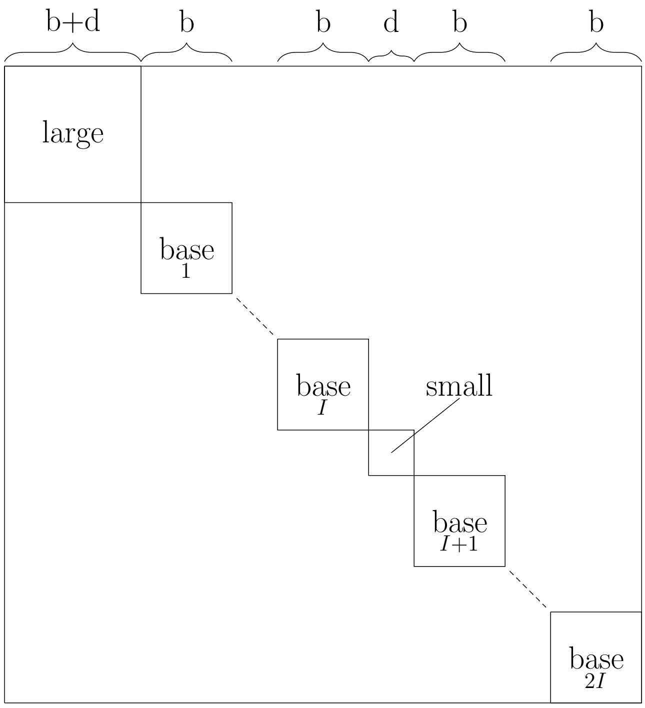

Figure 2.1: The positioning of the subsquares. By taking the union of partial transversals of B_n in these subsquares, we find a transversal of B_n.

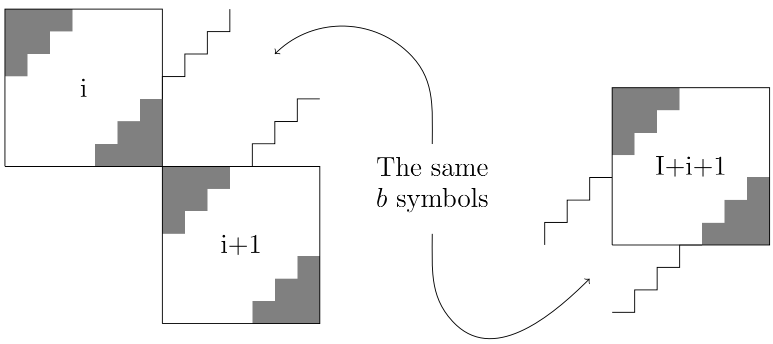

Figure 2.2: We choose partial transversal such that the symbols not used between the ith and $(i+1)$th base subsquares are used in the $(I+i+1)$th base subsquare. The darkened cells represent those cells that we do not allow in any partial transversal.

$t_0 \in \Omega_\mu^b(b+d)$ and $t_i \in \Omega_\mu^b(b)$, for $1 \le i \le 2I$.

We provide the following construction, followed by a proof that demonstrates that the construction yields Theorem 2.2.6.

Construction 2.2.7. *Take $\mu \ge 2$ an integer and n, b odd integers, $3 \le b \le n/3$ Let $I \ge 1$ and $d \in [0, b-1]$ be the unique integers such that $n = 2bI + b + 2d$.*

We will construct μ subsets of B_n, $T_1, \ldots, T_\mu$ by finding partial transversals selected from a large subsquare $B_{0,b+d}$, a small subsquare $B_{b(I+1)+d,d} = B_{(n+b)/2,d}$, and base subsquares $B_{bi+d,b}$ and $B_{b(I+i)+2d,b}$ for $1 \le i \le I$.

For the large subsquare, as $t_0 \in \Omega_\mu^b(b+d)$ there exists a collection of μ partial transversals $P_1^L, \ldots, P_\mu^L$ within $B_{0,b+d}$ that intersects stably in t_0 points and using each symbol of $\{(b-1)/2, \ldots, 2d+3(b-1)/2\} \setminus \{b+2d' \mid 0 \le d' < d\}$ precisely once per partial transversal. We place the cells of P_β^L into T_β, $1 \le \beta \le \mu$.

For the small subsquare, a collection of μ partial transversals $P_1^S, \ldots, P_\mu^S$ within $B_{b(I+1)+d,d}$ that intersect stably in d points can be defined by placing cells $(r, r, 2r)$ with $r = b(I+1) + d + d'$ into every partial transversal P_β^S, $1 \le \beta \le \mu$ and $0 \le d' < d$, so that each of the μ partial transversals are identical. We place the cells of P_β^S into T_β, $1 \le \beta \le \mu$.

For the first set of base subsquares, $B_{bi+d,b}$ with $1 \le i \le I$, as $t_i \in \Omega_\mu^b(b)$ there exists a collection of μ partial transversals $P_1^i, \ldots, P_\mu^i$ of $B_{0,b}$ that intersect stably in t_i points and using each symbol of $\{(b-1)/2, \ldots, 3(b-1)/2\}$ precisely once. For every $(r, c, e) \in P_\beta^i$, place the cells $(r+a, c+a, e+2a)$ with $a = bi + d$ into T_β, $1 \le \beta \le \mu$. The cells that were just filled are in the subsquare $B_{bi+d,b}$.

For the second set of base subsquares, $B_{b(I+i)+2d,b}$ with $1 \le i \le I$, as $t_{I+i} \in \Omega_\mu^b(b)$ there exists a collection of μ partial transversals $P_1^{I+i}, \ldots, P_\mu^{I+i}$ of $B_{0,b}$ that intersect stably in t_{I+i} points and using each symbol of $\{(b-1)/2, \ldots, 3(b-1)/2\}$ precisely once. For every

$(r, c, e) \in P_\beta^{I+i}$, place the cells $(r+a, c+a, e+2a)$ with $a = b(I+i)+2d$ into T_β, $1 \le \beta \le \mu$. The cells that were just filled are in the subsquare $B_{b(I+i)+2d,b}$.

Proof. We begin by showing that $T_1, \ldots, T_\mu$ from Construction 2.2.7 are each diagonals. Consider any $T \in \{T_1, \ldots, T_\mu\}$. As T is the union of partial transversals of subsquares, each of which share no common row or column, clearly T is a selection of n cells of L using each row (resp. column) once, and so T is a diagonal of B_n.

We will proceed to show that each diagonal $T \in \{T_1, \ldots, T_\mu\}$ is a transversal, and that they intersect stably in $d + \sum_{i=0}^{2I} t_i$ points. The construction placed $2b$ filled cells from the two base subsquares $B_{bi+d,b}$ and $B_{b(I+i)+2d,b}$ into T, for each fixed i, $1 \le i \le I$. This consisted of precisely one filled cell for each symbol of $\{2bi+2d-(b+1)/2, \ldots, 2bi+2d+3(b-1)/2\}$.

Then collectively the $2I$ base subsquares were used to fill $2bI$ cells into T, placing precisely one filled cell for each symbol of $\{2d + 3(b-1)/2 + 1, \ldots, 2bI + 2d + 3(b-1)/2\}$.

During the construction, $(b+d)+d$ filled cells were placed into T from the large subsquare $B_{0,b+d}$ and the small subsquare $B_{b(I+1)+d,d}$, which had one filled cell for each symbol of $\{(b-1)/2, \ldots, 2d + 3(b-1)/2\}$.

Combining the statements for the $2I$ base subsquares and the large and small subsquare, each symbol of $\{(b-1)/2, \ldots, 2bI+2d+3(b-1)/2\} = \{0, \ldots, 2bI+b+2d-1\}$ appears in the diagonal T precisely once, after recalling that each symbol is taken modulo $2bI+b+2d$ and noting that $2bI + 2d + 3(b-1)/2 = (2bI + b + 2d) + (b-1)/2 - 1 \equiv (b-1)/2 - 1 \pmod{n}$.

This shows that T is indeed a transversal, and so the construction has indeed formed μ transversals. Now we need to show that the μ transversals intersect stably in $d + \sum_{i=0}^{2I} t_i$ points. Suppose the μ partial transversals we chose for the large subsquare intersect stably in the set S_0, the μ partial transversals we chose for the base subsquare $B_{bi+d,b}$

intersect stably in the set S_i, and the μ partial transversals we chose for the base subsquare $B_{b(I+i)+2d,b}$ intersect stably in the set S_{I+i}, for $1 \leq i \leq I$. Clearly the μ partial transversals we chose for the small subsquare intersect stably in the points $S_{-1} = \{(r, r, 2r) \mid r = b(I+1)+d+d' \text{ and } 1 \leq d' \leq d\}$. The size of S_i is $|S_i| = t_i$, for $0 \leq i \leq 2I$, and $|S_{-1}| = d$. The μ transversals then clearly intersect stably in the $d + \sum_{i=0}^{2I} t_i$ points $\bigcup_{i=-1}^{2I} S_i$. $\square$

Example 2.2.8. *We consider the case when* $\mu = 2$, $n = 17$, $b = 5$, $d = 1$, $I = 1$, $t_0 = 1$, $t_1 = 1$, $t_2 = 0$. *Note that* $n = 2bI + b + 2d = 5 \cdot 2 \cdot 1 + 5 + 2 \cdot 1 = 17$.

For this example, we will represent the first transversal of $B_{i,j}$ *or* B_n *by underlining those entries, and the second transversal by adding a superscripted star. The intersection of the two (partial) transversals are those entries that are both underlined and starred.*

For the small subsquare, we require $\mu = 2$ *transversals of the small subsquare* $B_{(I+1)b+d,d} = B_{11,1}$ *that intersect stably in* $d = 1$ *points. These transversals are simply chosen as there is only one cell in* $B_{11,1}$.

For the large subsquare, we require $\mu = 2$ *transversals of the large subsquare* $B_{0,b+d} = B_{0,6}$ *that intersect stably in* $t_0 = 1$ *points using symbols* $\{(b-1)/2, \ldots, 2d+3(b-1)/2\} \setminus \{b+2d' \mid 0 \leq d' < d\} = \{2, 3, 4, 6, 7, 8\}$, *for example:*

0	1	2	3^*	$\underline{4}$	5
1	2	$\underline{3}$	4	5	6^*
$\underline{2}^*$	3	4	5	6	7
3	4^*	5	6	7	$\underline{8}$
4	5	6	$\underline{7}$	8^*	9
5	$\underline{6}$	7^*	8	9	10

For the first set of base subsquares (in this case the set contains only one subsquare) we require $\mu = 2$ *transversals of the base subsquare* $B_{0,b} = B_{0,5}$ *that intersect stably in* $t_1 = 1$ *points and using symbols* $\{(b-1)/2, \ldots, 3(b-1)/2\} = \{2, 3, 4, 5, 6\}$, *for example:*

0	1	2	3^*	$\underline{4}$
1	2	$\underline{3}$	4	5^*
$\underline{2}^*$	3	4	5	6
3	4^*	5	$\underline{6}$	7
4	$\underline{5}$	6^*	7	8

For the second set of base subsquares (in this case the set contains only one subsquare), we require $\mu = 2$ transversals of the base subsquare $B_{0,b} = B_{0,5}$ that intersect stably in $t_2 = 0$ points, for example:

0	1	2	3^*	$\underline{4}$
1	2^*	$\underline{3}$	4	5
$\underline{2}$	3	4	5	6^*
3	4	5^*	$\underline{6}$	7
4^*	$\underline{5}$	6	7	8

Then we can obtain $\mu = 2$ transversals of size $n = 17$ and stable intersection size $d + t_0 + t_1 + t_2 = 3$ as (where we omit those entries not relevant to our construction):

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
0	0	1	2	3^*	$\underline{4}$	5											
1	1	2	$\underline{3}$	4	5	6^*											
2	$\underline{2^*}$	3	4	5	6	7											
3	3	4^*	5	6	7	$\underline{8}$											
4	4	5	6	$\underline{7}$	8^*	9											
5	5	$\underline{6}$	7^*	8	9	10											
6							12	13	14	15^*	$\underline{16}$						
7							13	14	$\underline{15}$	16	0^*						
8							$\underline{14^*}$	15	16	0	1						
9							15	16^*	0	$\underline{1}$	2						
10							16	$\underline{0}$	1^*	2	3						
11												$\underline{5^*}$					
12													7	8	9	10^*	$\underline{11}$
13													8	9^*	$\underline{10}$	11	12
14													$\underline{9}$	10	11	12	13^*
15													10	11	12^*	$\underline{13}$	14
16													11^*	$\underline{12}$	13	14	15

This concludes the example.

2.3 Application to $\mu = 3, 4$

Our approach to finding 3 (resp. 4) transversals of B_n that intersect stably is to find 3 (resp. 4) partial transversals of $B_{i,j}$ that intersect stably for certain values of i and j, and compose these into transversals of B_n. We will use Theorem 2.2.6 with base sizes of $b = 9, 11, 15$. These sizes have been chosen based upon the results of a computational search for partial transversals of base and large subsquares.

The appendix includes tables that contains a set of either three or four rows, corresponding to $\mu = 3$ and $\mu = 4$ respectively, each row containing $b + d$ symbols. For the rth row, denote the ith symbol of the list in this row as a_i^r, for $1 \leq r \leq \mu$ and $1 \leq i \leq b+d$. The cells (i, a_i^r), $1 \leq i \leq b+d$, form a partial transversal of $B_{0,b+d}$. The three (resp. four) rows give three (resp. four) partial transversals of $B_{0,b+d}$, each partial transversal having t cells that are common amongst all three (resp. four) partial transversals, and $b+d-t$ cells which do not appear in the other partial transversals. We call this representation a *reduced form*.

We take the addition and scalar multiplication of finite sets to be:

$$A + B = \{a + b \mid a \in A, b \in B\}$$
$$kA = \{\sum_{i=1}^{k} a_i \mid a_i \in A\}$$

Lemma 2.3.1. *Let j, a, b be positive integers with $1 \leq a < b$. We have $j([0,a] \cup \{b\}) = [0, jb] \setminus \bigcup_{i=1}^{\lfloor (b-2)/a \rfloor} [jb - ib + ia + 1, jb - ib + b - 1]$.*

Proof. From definition, $j([0,a] \cup \{b\}) = \{\sum_{i=1}^{j} a_i \mid a_i \in [0,a] \cup \{b\}\} = \bigcup_{i=0}^{j} [i \cdot b, i \cdot b + (j-i)a] = \bigcup_{i=0}^{j} [(j-i)b, (j-i)b + ia]$. Then any value $t \in [0, jb]$ with $t \notin j([0,a] \cup \{b\})$ must be between the two intervals $[(j-i')b, (j-i')b + i'a]$ and $[(j-i'+1)b, (j-i'+1)b + (i'-1)a]$ for some $1 \leq i' \leq j$, and hence $t \in [(j-i')b + i'a + 1, (j-i'+1)b - 1]$. This proves the result, once we note that $[(j-i')b + i'a + 1, (j-i'+1)b - 1]$ is non-empty only when $i'a + 1 \leq b - 1$, and so $i' \leq (b-2)/a$. □

2.3.1 Existence of partial transversals in subsquares

It is important to note that $\Omega_4^b(b+d) \subseteq \Omega_3^b(b+d)$. Also, if there is at least one partial transversal of $B_{0,b+d}$ using symbols $\{(b-1)/2, \ldots, 2d + 3(b-1)/2\} \setminus \{b + 2d' \mid 0 \leq d' < d\}$ then $b + d \in \Omega_\mu^b(b+d)$ for any $\mu \geq 2$. This also tells us that if $\Omega_\mu^b(b+d) \neq \emptyset$, then $b + d \in \Omega_{\mu'}^b(b+d)$ for each $\mu' \geq \mu$.

Lemma 2.3.2. *The following hold*[1]*:*

1. $\Omega_4^9(9) \supseteq \{0, 1, 9\}$.

2. $\Omega_4^{11}(11) \supseteq \{0, 1, 2, 3, 11\}$.

3. $\Omega_4^{15}(15) \supseteq \{1, 2, 3, 4, 5, 15\}$.

4. $\Omega_3^9(9) \supseteq \{0, 1, 2, 3, 9\}$.

5. $\Omega_3^{11}(11) \supseteq \{0, 1, 2, 3, 4, 5, 11\}$.

Proof. The corresponding partial transversals have been found by a computer search, and have been written in reduced form in the appendix, in respectively Table A.1 and A.2, Table A.5, Table A.9, Tables A.1, A.2, and A.4, and Tables A.5 and A.8. □

Lemma 2.3.3. *The following hold:*

1. $0 \in \Omega_4^9(9 + d)$, *for all* $0 \leq d < 9$.

2. $3 \in \Omega_4^{11}(11 + d)$, *for all* $0 \leq d < 11$.

Proof. The corresponding partial transversals have been found by a computer search, and have been written in reduced form in the appendix, in respectively Tables A.1, A.2, and A.3 , and Tables A.5, A.6, and A.7. □

Lemma 2.3.4. *The following hold:*

1. $11 + d \in \Omega_4^{11}(11 + d)$, *for all* $0 \leq d < 11$.

2. $15 + d \in \Omega_4^{15}(15 + d)$, *for all* $0 \leq d < 15$.

[1]In each case equality holds, but this strengthened statement is not needed.

Proof. Since the partial transversals required intersect stably in the same number of points as the square size, we only need one partial transversal of $B_{0,b+d}$, which is repeated 4 times to form the 4 partial transversals that intersect stably in $b+d$ points. One partial transversal has been been found for each of the 26 cases by a computer search, and these have been written in reduced form in the appendix, in respectively Tables A.5, A.6 and A.7, and Table A.10. □

Lemma 2.3.5. *The following set relations hold:*

1. $2I\Omega_4^9(9) = [0, 18I] \setminus \cup_{i=1}^{7}[18I - 8i + 1, 18I - 9i + 8]$;
2. $2I\Omega_4^{11}(11) = [0, 22I] \setminus (\{22I - 23\} \cup [22I - 15, 22I - 12] \cup [22I - 7, 22I - 1])$;
3. $2I\Omega_4^{15}(15) = [2I, 30I] \setminus (\{30I - 29\} \cup [30I - 19, 30I - 15] \cup [30I - 9, 30I - 1])$;
4. $2I\Omega_3^9(9) = [0, 18I] \setminus (\{18I - 11, 18I - 10\} \setminus [18I - 5, 18I - 1])$; *and*
5. $2I\Omega_3^{11}(11) = [0, 22I] \setminus [22I - 5, 22I - 1]\}$.

Proof. The sets $\Omega_\mu^9(9)$, $\Omega_\mu^{11}(11)$ and $\Omega_4^{15}(15) = \{1\} + \{0, 1, 2, 3, 4, 14\}$ are given in Lemma 2.3.2 for $\mu = 3, 4$. For the case $\Omega_\mu^9(9)$ and $\Omega_\mu^{11}(11)$, Lemma 2.3.1 completes the result for general J, however we will only be requiring the case when J is even, and hence written $J = 2I$. For the case $\Omega_4^{15}(15)$, it can be seen that $2I\Omega_4^{15}(15) = 2I\{1, 2, 3, 4, 5, 15\} = 2I(\{1\} + \{0, 1, 2, 3, 4, 14\}) = \{2I\} + 2I\{0, 1, 2, 3, 4, 14\}$. We can apply Lemma 2.3.1 to find $2I\{0, 1, 2, 3, 4, 14\}$, which gives the final result. □

2.3.2 $\mu = 3$

Theorem 2.3.6. *For odd $n \geq 33$, let I' and d' be the unique integers such that $n = 22I' + 11 + 2d'$, $I' \geq 1$ and $0 \leq d' < 11$. Then there exist three transversals of B_n that intersect stably in t points for $t \in [11 + 2d', n] \setminus [n - 5, n - 1]$.*

Proof. Take the base size to be $b = 11$. Using Theorem 2.2.6, Lemma 2.3.4, and Lemma 2.3.5, we can conclude there exists the required collection of transversals for each $t \in \{d'\} + \Omega_3^{11}(11 + d') + 2I'\Omega_3^{11}(11)$, and hence for each $t \in \{d'\} + \{11 + d'\} + 2I'\Omega_3^{11}(11) = [11 + 2d', n] \setminus [n - 5, n - 1]$. $\square$

Lemma 2.3.7. *For odd $n \geq 27$, let I and d be the unique integers such that $n = 18I + 9 + 2d$, $I \geq 1$ and $0 \leq d < 9$. Then there exist three transversals of B_n that intersect stably in t points for $t \in [d, 18I + d] \setminus ([18I - 11 + d, 18I - 10 + d] \cup [18I - 5 + d, 18I - 1 + d])$.*

Proof. Take the base size to be $b = 9$. Using Theorem 2.2.6, Lemma 2.3.3, and Lemma 2.3.5, we can conclude there exists the required collection of transversals for each $t \in \{d\} + \{0\} + 2I\Omega_3^9(9) \subseteq \{d\} + \Omega_3^9(9 + d) + 2I\Omega_3^9(9)$, and hence for each $t \in [d, 18I + d] \setminus ([18I - 11 + d, 18I - 10 + d] \cup [18I - 5 + d, 18I - 1 + d])$. $\square$

Lemma 2.3.8. *For odd $n \geq 33$, let I' and d' be the unique integers such that $n = 22I' + 11 + 2d'$, $I' \geq 1$ and $0 \leq d' < 11$. Then there exist three transversals of B_n that intersect stably in t points for $t \in [3 + d', 22I' + 3 + d'] \setminus [22I' - 2 + d', 22I' + 2 + d']$.*

Proof. Take the base size to be $b = 11$. Using Theorem 2.2.6, Lemma 2.3.3, and Lemma 2.3.5, we can conclude there exists the required collection of transversals for each $t \in \{d'\} + \{3\} + 2I\Omega_3^{11}(11) \subseteq \{d'\} + \Omega_3^{11}(11 + d') + 2I\Omega_3^{11}(11)$, and hence for each $t \in [3 + d', 22I' + 3 + d'] \setminus [22I' - 2 + d', 22I + 2 + d']$. $\square$

Theorem 2.3.9. *For odd $n \geq 33$, let I, I', d and d' be the unique integers such that $n = 18I + 9 + 2d$ and $n = 22I' + 11 + 2d'$, $I, I' \geq 1$, $0 \leq d < 9$ and $0 \leq d' < 11$. Then there exist three transversals of B_n that intersect stably in t points for $t \in [\min(3 + d', d), 11 + 2d']$ except, perhaps, when:*

- *$n = 51$ and $t = 29$,*
- *$n = 53$ and $t = 30$.*

Proof. We first show that we have three transversals of B_n that intersect stably in t points for $t \in [3+d', 11+2d']$, except in the case $n = 51$ and $t = 29$, and the case $n = 53$ and $t = 30, 31$. Lemma 2.3.8 gives the cases when $t \in [3+d', 22I'-3+d']$. Now if $11+2d' \leq 22I'-3+d'$, then we are done. Otherwise $d' > 22I'-14$, and since $d' \leq 10$, this implies $I' = 1$ and $d' \in \{9, 10\}$. The case $d' = 9$ gives $n = 51$, and we do not have three transversals of B_n that intersect stably in t when $t \in [22I'-2+d', 11+2d'] = \{29\}$. The case $d' = 10$ gives $n = 53$, and we do not have three transversals of B_n that intersect stably in t when $t \in [22I'-2+d', 11+2d'] = \{30, 31\}$. We note that the case $n = 53$ and $t = 31$ is covered by Lemma 2.3.7.

Second we show that we have those cases with $t \in [d, 3+d']$ when $d < 3+d'$. For $33 \leq n \leq 43$, $d = 3+d'$, so assume $n \geq 45$, implying $I \geq 2$. By Lemma 2.3.7, we have those cases with $t \in [d, 18I-12+d]$, and since $18I-12+d \geq 24 > 3+d'$, we are done. □

Then Theorem 2.1.3 follows by Theorem 2.3.6 and Theorem 2.3.9.

2.3.3 $\mu = 4$

Lemma 2.3.10. *For odd $n \geq 33$, let I' and d' be the unique integers such that $n = 22I'+11+2d'$, $I' \geq 1$ and $0 \leq d' < 11$. Then there exist four transversals of B_n that intersect stably in t points for $t \in [11+2d', n] \setminus \{n-23, n-15, \ldots, n-12, n-7, \ldots, n-1\}$.*

Proof. Take the base size to be $b = 11$. Using Theorem 2.2.6, Lemma 2.3.4, and Lemma 2.3.5, we can conclude there exists the required collection of transversals for each $t \in \{d'\} + \Omega_4^{11}(11+d') + 2I'\Omega_4^{11}(11)$, and hence for each $t \in \{11+2d'\} + 2I'\Omega_4^{11}(11) = [11+2d', n] \setminus (\{n-23\} \cup [n-15, n-12] \cup [n-7, n-1])$. □

Lemma 2.3.11. *For odd $n \geq 45$, let I'' and d'' be the unique integers such that $n = 30I''+15+2d''$, $I'' \geq 1$ and $0 \leq d'' < 15$. Then there exist four transversals of B_n that* *intersect stably in t points for $t \in [2I''+15+2d'', n] \setminus (\{n-29\} \cup [n-19, n-15] \cup [n-9, n-1])$.*

Proof. Using Theorem 2.2.6, Lemma 2.3.4, and Lemma 2.3.5, we can conclude there exists the required collection of transversals for each $t \in \{d''\} + \Omega_4^{15}(15+d'') + 2I''\Omega_4^{15}(15)$, and hence for each $t \in \{15+2d''\} + 2I''\Omega_4^{15}(15) = [2I''+15+2d'', n] \setminus (\{n-29\} \cup [n-19, n-15] \cup [n-9, n-1])$. □

Theorem 2.3.12. *For odd $n \geq 45$, let I' and d' be the unique integers such that $n = 22I'+11+2d'$, $I' \geq 1$, $0 \leq d' < 11$. Then there exist four transversals of B_n that intersect stably in t points for $t \in [11+2d', n] \setminus (\{n-15\} \cup [n-7, n-1])$. For odd $33 \leq n \leq 43$ such that $n = 33+2d'$ and $0 \leq d' \leq 5$, there exists four transversal of B_n that intersect stably in t points, for $t \in [11+2d', n] \setminus ([n-15, n-12] \cup [n-7, n-1])$.*

Proof. Define I'' and d'' such that $n = 30I''+15+2d''$, $I'' \geq 1$ and $0 \leq d'' < 15$. This theorem is the union of the result from Lemma 2.3.10 and Lemma 2.3.11. The case for $I'' \geq 1$ requires the knowledge that $\{n-23\} \cup [n-14, n-12] \subseteq [2I''+15+2d'', n] \setminus (\{n-29\} \cup [n-19, n-15] \cup [n-9, n-1])$, which is easily seen as $2I''+15+2d'' \leq n-23 = 30I''+15+2d''-23$ when $I'' \geq 1$. Then the union of $[11+2d', n] \setminus (\{n-23\} \cup [n-15, n-12] \cup [n-7, n-1])$ and $[2I''+15+2d'', n] \setminus (\{n-29\} \cup [n-19, n-15] \cup [n-9, n-1])$ gives the result as stated in the theorem when $n \geq 45$. The case for $33 \leq n < 45$ is covered by Lemma 2.3.10. □

Lemma 2.3.13. *For odd $n \geq 33$, let I' and d' be the unique integers such that $n = 22I'+11+2d'$, $I' \geq 1$ and $0 \leq d' < 11$. Then there exist four transversals of B_n that intersect stably in t points for $t \in [3+d', 22I'+3+d'] \setminus (\{22I'-20+d'\} \cup [22I'-12+d', 22I'-9+d'] \cup [22I'-4+d', 22I'+2+d'])$.*

Proof. Using Theorem 2.2.6, Lemma 2.3.3, and Lemma 2.3.5, we can conclude there exists the required collection of transversals for each $t \in \{d'\} + \{3\} + 2I'\Omega_4^{11}(11) \subseteq \{d\} + \Omega_4^{11}(11+d') + 2I'\Omega_4^{11}(11)$, and hence for each $t \in \{3+d', \ldots, 22I'+3+d'\} \setminus \{22I'-20+d', 22I'-12+d', \ldots, 22I'-9+d', 22I'-4+d', \ldots, 22I'+2+d'\}$. □

Lemma 2.3.14. *For odd $n \geq 27$, let I and d be the unique integers such that $n = 18I + 9 + 2d$, $I \geq 1$ and $0 \leq d < 9$. Then there exist four transversals of B_n that intersect stably in t points for $t \in [d, \ldots, 18I + d] \setminus \bigcup_{i=1}^{7} [18I - 8i + 1 + d, 18I - 9i + 8 + d]$.*

Proof. Using Theorem 2.2.6, Lemma 2.3.3, and Lemma 2.3.5, we can conclude there exists the required collection of transversals for each $t \in \{d\} + \{0\} + 2I\Omega_4^9(9) \subseteq \{d\} + \Omega_4^9(9+d) + 2I\Omega_4^9(9)$, and hence for each $t \in [d, 18I + d] \setminus \bigcup_{i=1}^{7} [18I - 8i + 1 + d, 18I - 9i + 8 + d]$. $\square$

Theorem 2.3.15. *For odd $n \geq 33$, let I, I', d and d' be the unique integers such that $n = 18I + 9 + 2d$ and $n = 22I' + 11 + 2d'$, $I, I' \geq 1$, $0 \leq d < 9$ and $0 \leq d' < 11$. Then there exist four transversals of B_n that intersect stably in t points for $t \in [\min(3+d', d), 11+2d']$ except, perhaps, when:*

- *$33 \leq n \leq 43$ and $t \in [10 + d', 11 + d']$,*
- *$45 \leq n \leq 53$ and $t \in [-1 + d', 2 + d'] \cup [10 + d', 11 + d'] \cup [18 + d', 20 + d']$,*
- *$63 \leq n \leq 75$ and $t \in [7 + d, 8 + d]$.*

Proof. For $33 \leq n \leq 43$, we have $I' = I = 1$ and $\min(3 + d', d) = 3 + d' = d$. Then we have the existence of four transversal of B_n that intersect stably in t points by Lemma 2.3.13 for $t \in [3+d', 25+d'] \setminus ([10+d', 13+d'] \cup [18+d', 24+d'])$ and by Lemma 2.3.14 for $t \in [d, 18+d] \setminus ([3+d, 8+d] \cup [11+d, 17+d]) = [3+d', 21+d'] \setminus ([6+d', 11+d'] \cup [14+d', 20+d'])$. The union of the two result sets is $[3 + d', 25 + d'] \setminus ([10 + d', 11 + d'] \cup [18 + d', 20 + d'] \cup [22 + d', 24 + d'])$. A subset of this is $[3 + d', 16 + d'] \setminus [10 + d', 11 + d']$. Noting that as $d' \leq 5$ for the specified n, then $11 + 2d' \leq 16 + d'$, and so this subset includes the range $[\min(3 + d', d), 11 + 2d'] \setminus [10 + d', 11 + d']$, which is the required result when $33 \leq n \leq 43$.

For $45 \leq n \leq 53$, we have $I = 2$, $I' = 1$ and as $d' = 6 + d$ we have $\min(3 + d', d) = d$. Then we have the existence of four transversal of B_n that intersect stably in t points by Lemma

2.3.13 for $t \in [3+d', 25+d'] \setminus ([10+d', 13+d'] \cup [18+d', 24+d'])$ and by Lemma 2.3.14 for $[d, 36+d] \setminus ([5+d, 8+d] \cup [13+d, 17+d] \cup [21+d, 26+d] \cup [29+d, 35+d]) = [-6+d', 30+d'] \setminus ([-1+d', 2+d'] \cup [7+d', 11+d'] \cup [15+d', 20+d'] \cup [23+d', 29+d'])$. The union of the two result sets is $[d, \ldots, 30+d'] \setminus ([-1+d', 2+d'] \cup [10+d', 11+d'] \cup [18+d', 20+d'] \cup [23+d', 24+d'] \cup [26+d', 29+d'])$. A subset of this is $[d, 21+d'] \setminus ([-1+d', 2+d'] \cup [10+d', 11+d'] \cup [18+d', 20+d'])$. Noting that as $d' \leq 10$, then $11 + 2d' \leq 21 + d'$, and so this subset includes the range $[\min(3 + d', d), 11 + 2d'] \setminus ([-1 + d', 2 + d'] \cup [10 + d', 11 + d'] \cup [18 + d', 20 + d'])$, which is the required result when $45 \leq n \leq 53$.

For $n \geq 55$, we have $I' \geq 2$. Then we have the existence of four transversal of B_n that intersect stably in t points by Lemma 2.3.13 for $t \in [3 + d', 21 + d']$. This completes the case when $\min(3 + d', d) = 3 + d'$, as $11 + 2d' \leq 21 + d'$. When $d < 3 + d'$, we still need the cases $t \in [d, \ldots, 3 + d']$. As $3 + d' \leq 13$, it is enough to show the statement holds for those t with $t \in [d, 13]$.

When $I = 3$, then $63 \leq n \leq 75$, and we have the existence of four transversal of B_n that intersect stably in t points by Lemma 2.3.14 for $t \in [d, 13] \setminus [7 + d, 8 + d]$. When $I \geq 4$, we have the existence of four transversal of B_n that intersect stably in t points by Lemma 2.3.14 for $t \in [d, 13]$. □

Then Theorem 2.1.4 follows by Theorem 2.3.12 and Theorem 2.3.15.

2.4 Application to latin trades

Let D represent a combinatorial design and assume there exists distinct sets S_1, S_2 with $S_1 \subseteq D$, such that $D' = (D \setminus S_1) \cup S_2$ forms a valid design. Then the pair (S_1, S_2) forms a *combinatorial bitrade*. The original design D is immaterial, and we can define a bitrade formally by taking the pair of sub-designs (S_1, S_2) that fulfill certain properties. If our combinatorial design is a latin square, the bitrade is called a latin bitrade. A good survey of latin bitrades is [23], and for trades in general is [11].

Definition 2.4.1. *A* μ-way latin trade *of volume* s *and order* n *is a collection of* μ *partial latin squares* $(L_1, \ldots, L_\mu)$*, each of order* n*, such that:*

1. *Each partial latin square contains exactly the same* s *filled cells,*
2. *If cell* (i, j) *is filled then it contains a different entry in each of the* μ *partial latin squares,*
3. *Row* i *in each of the* μ *partial latin squares contains, set-wise, the same symbols, and column* j *likewise.*

A μ-way latin trade is *circulant* if each of the partial latin squares can be obtained from the first row by simultaneously cycling the rows, columns, and symbols. For example, the cell $(r, c, e) \in L$ would imply $(r + 1, c + 1, e + 1) \in L$. We call the set of first rows the *base row*, and can write it in the notation $B = \{(e_1, \ldots, e_\mu)_{c_j} \mid 1 \leq j \leq k\}$, where $(0, c_j, e_\alpha) \in L_\alpha$ for $1 \leq \alpha \leq \mu$.

A μ-way latin trade is *k-homogeneous* if in each partial latin square, L, each row and each column contain k filled cells, and each symbol appears in filled cells of L precisely k times. Clearly a circulant μ-way trade is k-homogeneous, where k is the number of filled cells in the first row.

There has been much interest in 2-way k-homogeneous latin trades as demonstrated by the work in [8], [10], [20], [24], [25], and [59], and more recently there has been an extension to μ-way k-homogeneous latin trades in [7].

Theorem 2.4.2. *If there exists a collection of* μ *transversals of* B_n *that intersect stably in* t *points, then there exists a circulant* μ*-way* $(n - t)$*-homogeneous latin trade of order* n*.*

Proof. Consider a collection of μ transversals of B_n, $T_1, \ldots, T_\mu$, that intersect stably in the t points S. Consider the partial latin squares $Q_\alpha = \{(i, c+i, r+c+i) \mid 0 \leq i \leq n-1 \text{ and } (r, c, r+c) \in T_\alpha \setminus S\}$. It is clear that each corresponding row of the Q_α contain setwise the same symbols. As the cells of the first column of Q_α are $(-c, 0, r) \in Q_\alpha$, each column contain setwise the same symbols. Then it is clear that the collection of μ partial latin squares satisfy the conditions of a μ-way latin trade. They are also circulant by definition, and hence are clearly $(n-t)$-homogeneous. □

Example 2.4.3. *Consider B_5 with the following transversals:*

$\underline{0}^*$	1	2	3	4
1	$\underline{2}$	3^*	4	0
2	3	$\underline{4}$	0	1^*
3	4^*	0	$\underline{1}$	2
4	0	1	2^*	$\underline{3}$

Here, the transversals intersect stably in the 1 *point* $S = \{(0, 0, 0)\}$. *The cell* $(1, 2, 3)$ *is in the starred transversal, and not in* S, *so Construction 2.4.2 places the cell* $(0, 2, 3)$ *into the resulting first row of a circulant latin square. Construction 2.4.2 gives the first row of a circulant latin squares to be:*

·	4	3	2	1

·	2	4	1	3

Writing these latin squares out completely:

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

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

The two partial latin squares form a 2-way 4-homogeneous circulant latin trade of order 5. This completes the example.

Theorem 2.4.4. *For odd $n \geq 33$, let I, I', d and d' be the unique integers such that $n = 18I + 9 + 2d$ and $n = 22I' + 11 + 2d'$, $I, I' \geq 1$, $0 \leq d < 9$ and $0 \leq d' < 11$. Then there exists a circulant $(n-t)$-homogeneous 3-way latin trade of order n, for $t \in [\min(3+d', d), n] \setminus [n-5, n-1]$, except, perhaps, when:*

- $n = 51$ *and* $t = 29$,
- $n = 53$ *and* $t = 30$.

Proof. Follows by Theorem 2.1.3 and Theorem 2.4.2. □

Theorem 2.4.5. *For odd $n \geq 33$, let I, I', d and d' be the unique integers such that $n = 18I + 9 + 2d$ and $n = 22I' + 11 + 2d'$, $I, I' \geq 1$, $0 \leq d < 9$ and $0 \leq d' < 11$. Then there exists a circulant $(n-t)$-homogeneous 4-way latin trade of order n, for $t \in [\min(3+d', d), n] \setminus (\{n-15\} \cup [n-7, n-1])$, except, perhaps, when:*

- $33 \leq n \leq 43$ *and* $t \in [10+d', 11+d'] \cup [n-14, n-12]$,
- $45 \leq n \leq 53$ *and* $t \in [-1+d', 2+d'] \cup [10+d', 11+d'] \cup [18+d', 20+d']$,
- $63 \leq n \leq 75$ *and* $t \in [7+d, 8+d]$.

Proof. Follows by Theorem 2.1.4 and Theorem 2.4.2. □

2.5 Conclusion and future work

We have been able to show, with a number of exceptions, that their exists three (resp. four) transversals of B_n that intersect stably in t points when n is odd and $n \geq 33$. With only a few unsolved cases, it appears that future work may be able to answer Question 1.3.1 completely for $\mu = 3, 4$.

Theorem 2.4.4 and 2.4.5 fill in a large portion of the spectrum of 3/4-way k-homogeneous latin trades of odd order, which is a significant advancement on what was previously known. There are a number of construction for μ-way k-homogeneous latin trades [7], and it seems that further work may result in the spectrum being completed.

Chapter 3

μ-way k-homogenous latin trades

3.1 Introduction

In this chapter, the symbols in a latin square are $\Omega = [m] = \{1, \ldots, m\}$. The rows and columns will be indexed by $\Omega = [m]$. For convenience, we restate the definitions that we will use the most in the upcoming chapter:

Definition 3.1.1. *For natural numbers μ, m, $\mu \leq m$, a* μ-way latin trade of order m on symbol set Ω *is a collection $\mathcal{T} = (T_1, \ldots, T_\mu)$ of μ partial latin squares of order m using symbols of Ω such that:*

- *$\mathcal{S}(T_\alpha) = \mathcal{S}(T_\beta)$, for each $1 \leq \alpha < \beta \leq \mu$;*
- *for each $(r, c) \in \mathcal{S}(T_\alpha)$ it holds that $t_\alpha(r, c) \neq t_\beta(r, c)$, for every $1 \leq \alpha < \beta \leq \mu$; and*
- *$\mathcal{R}_r(T_\alpha) = \mathcal{R}_r(T_\beta)$ and $\mathcal{C}_c(T_\alpha) = \mathcal{C}_c(T_\beta)$, for each $r, c \in [m]$ and $1 \leq \alpha < \beta \leq \mu$.*

Definition 3.1.2. *For an integer $k \geq 0$, a* (μ, k, m)-latin trade *on symbol set Ω, $\mathcal{T} = (T_1, \ldots, T_\mu)$, is a μ-way latin trade of order m on Ω that has $k = |\mathcal{R}_r(\mathcal{T})| = |\mathcal{C}_c(\mathcal{T})| = |\mathcal{E}_e(\mathcal{T})|$, for each $r, c, e \in [m]$. Such a μ-way latin trade is called* k-homogeneous.

A (μ, k, m)-latin trade can have $k = 0$ in the case that each of the μ partial latin squares is empty; otherwise k must satisfy $\mu \leq k \leq m$.

We will require the (μ, k, m)-latin trades that we investigate to have the property that if $(r, c, e) \in T_\alpha$, where T_α is one of the partial latin squares that forms the (μ, k, m)-latin trade, then r, c, e are pairwise distinct. With this property, $T_\alpha \cup \{(i, i, i) \mid i \in [m]\}$ would form a new partial latin square that resembles an idempotent latin square with some unfilled cells.

Definition 3.1.3. *A μ-way latin trade $\mathcal{T}$ of order m is* idempotent *if $i \notin \mathcal{R}_i(\mathcal{T}) \cup \mathcal{C}_i(\mathcal{T})$ and $(i, i) \notin \mathcal{S}(\mathcal{T})$, for $i \in [m]$.*

Definition 3.1.4. *A (μ, k, m)-latin trade is* circulant *if it can be obtained from the elements of its first row, called the base row (denoted by $\mu - B_m^k$), by simultaneously permuting each of the coordinates cyclically. That is, for each α, the cell $(1, c, e) \in T_\alpha$ implies $(1 + i, c + i \mod m, e + i \mod m) \in T_\alpha$, for $1 \leq i \leq m - 1$.*

We write the base row as $B = \{(a_1, \ldots, a_\mu)_{c_l} \mid 1 \leq l \leq k\}$, where $a_\alpha, c_l \in [m]$. Then the corresponding μ partial latin squares can be constructed as $T_\alpha = \{(1 + i, c_l + i \mod m, a_\alpha + i \mod m) \mid 0 \leq i \leq m - 1, 1 \leq l \leq k\}$, $\alpha \in [\mu]$. We will denote an idempotent circulant (μ, k, m)-latin trade by $\mu - IB_m^k$.

Let m be an integer. The spectrum of μ-way homogeneous latin trades of order m, $\mathcal{S}_m^\mu$, is the set of values of k such that there exists a (μ, k, m)-latin trade. The spectrum of idempotent μ-way homogeneous latin trades of order m, $\mathcal{IS}_m^\mu$, is the set of values of k such that there exists an idempotent (μ, k, m)-latin trade.

A previous study of (μ, k, m)-latin trades [7] posed the question:

Question 3.1.5. *For given m and k, $m \geq k \geq \mu$, does there exist a (μ, k, m)-latin trade?*

The primary goal of this chapter is to investigate this question by deducing $\mathcal{S}_m^3$. However in order to do this, we will use a construction that requires us to first investigate $\mathcal{IS}^3_m$. Clearly $\mathcal{IS}^3_m \subseteq \mathcal{S}^3_m$. It is known that $\{3,\ldots,m\} \supseteq \mathcal{S}^3_m$, and also that $3 \in \mathcal{S}^3_m$ if and only if $3|m$ (see [7]). In this chapter, we show there exists 3-way k-homogeneous latin trades of order m with $4 \leq k \leq m$ for all but a finite list of possible exceptions.

3.2 Literature review

A 2-way latin trade is typically called a latin bitrade. There have been three distinct approaches used to construct k-homogeneous latin bitrades.

The first approach used graph theoretic constructions (see also [20], [52], [53], and [59]):

Theorem 3.2.1. [24][25] *There exists a $(2, p, 3m)$-latin trade when $p = 3, 4$ and $m \geq 3$.*

The second approach used block theoretic based constructions:

Theorem 3.2.2. [8][10] *There exists a $(2, k, m)$-latin trade when $3 \leq k \leq 37$ and $m \geq k$.*

The third approach relies on finding pairs of transversals of given intersection in the back-circulant latin squares:

Theorem 3.2.3. [30] *For each odd $m \neq 5$ and for each $t \in \{0, \ldots, m-3\} \cup \{m\}$, there exists two transversals in B_m, T_1 and T_2, with $|T_1 \cap T_2| = t$. When $m = 5$ and for each $t \in \{0, 1, 5\}$, there exists two transversals in B_m, T_1 and T_2, with $|T_1 \cap T_2| = t$.*

Lemma 3.2.4. [30] *For m an odd integer, let T_1 and T_2 be two transversals in B_m such that $|T_1 \cap T_2| = t$. Then there exists a $(2, m-t, m)$-latin trade.*

These results lead to the completion of the spectrum problem for homogeneous latin bitrades:

Theorem 3.2.5. [8][10][24][25][30] *There is a* $(2, k, m)$*-latin trade for all* $3 \leq k \leq m$ *and a* $(2, 2, 2m)$*-latin trade, for all* $m \geq 1$.

The first study of (μ, k, m)-latin trades for general μ produced a number of block theoretic constructions [7] that yielded results for small k when $\mu = 3$:

Theorem 3.2.6. [7] *There exist* $(3, k, m)$*-latin trades for* $m \geq k$ *when:*

- $k = 3$ *and* $3 \mid m$*;*
- $k = 4$ *and* $m \neq 6, 7, 11$*;*
- $5 \leq k \leq 13$*;*
- $k = 15$*; and*
- $k = m$.

3.3 Idempotent generalization of basic constructions

The constructions that have appeared earlier in the literature for (μ, k, m)-latin trades [7] can be used (or modified) for the construction of idempotent (μ, k, m)-latin trades. As many of the constructions differ only trivially from their original appearance, we label the source of the original construction, and give the original proof with an extension if necessary.

Theorem 3.3.1. [7] *If there exist idempotent* (μ, k, m_i)*-latin trades, for* $i = 1, 2$*, then there exists an idempotent* $(\mu, k, m_1 + m_2)$*-latin trade.*

Theorem 3.3.2. [7] *If there exist an idempotent* (μ_1, k_1, m_1)*-latin trade and a* (μ_2, k_2, m_2)*-latin trade, then there exists an idempotent* $(\mu_1\mu_2, k_1k_2, m_1m_2)$*-latin trade.*

Theorem 3.3.3. [7] *If $l \neq 2, 6$ and for each $k \in \{k_2, ..., k_l\}$ there exists a (μ, k, p)-latin trade and there exists an idempotent (μ, k_1, p)-latin trade, then an idempotent $(\mu, k_1 + \cdots + k_l, lp)$-latin trade exists. (Some k_i's can possibly be zero.)*

Proof. For $l \neq 2, 6$, there exists two $l \times l$ orthogonal latin squares. Denote these latin squares by L_1 and L_2, with elements chosen from the sets $\{e_1, \ldots, e_l\}$ and $\{f_1, \ldots, f_l\}$, respectively. We can simultaneously permute the rows and columns of L_1 and L_2 so the main diagonal of L_2 contains only f_1, and then re-label the symbols of L_1 so that the symbols in cell (j, j) of L_1 is e_j. Assume that L^* is the square that is formed by superimposing L_1 and L_2. We replace each $(e_i, f_j) \in L^*$ such that $j \geq 2$ with a (μ, k_j, p)-latin trade whose elements are from the set $\{(i-1)p + 1, \ldots, ip\}$, and when $j = 1$ with an idempotent (μ, k_1, p)-latin trade whose elements are from the set $\{(i-1)p+1, \ldots, ip\}$. As a result we obtain a $(\mu, k_1 + \cdots + k_l, lp)$-latin trade, which we denote as $\mathcal{T}$.

Then clearly $(j, j) \notin \mathcal{S}(\mathcal{T})$ as each of the entries on the main diagonal of $\mathcal{T}$ came from an idempotent (μ, k_1, p)-latin trade. Note that in $\mathcal{T}$ a row $r \in \{(i-1)p+1, \ldots, ip\}$ contains cells filled with symbols $e \in \{(i-1)p+1, \ldots, ip\}$ only in columns $c \in \{(i-1)p+1, \ldots, ip\}$, and these filled cells came from an idempotent (μ, k, p)-latin trade. So if $(i-1)p + i' \in \mathcal{R}_{(i-1)p+i'}(\mathcal{T})$, $1 \leq i' \leq p$, then there must be a cell in row $(i-1)p + i'$ and column c with $c \in \{(i-1)p + 1, \ldots, ip\}$ that contains symbol $(i-1)p + i'$, and this comes from an idempotent (μ, k, m)-latin trade, say $\mathcal{U}$. But then $\mathcal{U}$ would have $i' \in \mathcal{R}_{i'}(\mathcal{U})$, a contradiction as $\mathcal{U}$ is idempotent. The analogous result holds for the columns, and $\mathcal{T}$ forms an idempotent $(\mu, k_1 + \cdots + k_l, lp)$-latin trade. □

Theorem 3.3.4. *Take k and k' to be integers with $k' > k$. If for every $k' \leq l \leq 2k' - 1$ there exists an idempotent (μ, k, l)-latin trade, then for any $m \geq k'$ there exists an idempotent (μ, k, m)-latin trade.*

Proof. For every $m \geq 2k'$, we can write $m = rk' + sl$, for some $r, s \geq 0$ and $k' + 1 \leq l \leq$

$2k' - 1$. Since there exist an idempotent (μ, k, k')-latin trade and an idempotent (μ, k, l)-latin trade, by Theorem 3.3.1 we conclude that there exists an idempotent (μ, k, m)-latin trade. $\square$

A large set of idempotent latin squares of order m is a set of $m-2$ idempotent latin squares of order m, $(L_1, \dots, L_{m-2})$, such that for α, β with $1 \leq \alpha < \beta \leq m-2$ and $i, j \in [m]$, $L_\alpha(i,i) = L_\beta(i,i) = i$ and $L_\alpha(i,j) \neq L_\beta(i,j)$ when $i \neq j$.

Theorem 3.3.5. *For $m \geq 3$, $m \neq 6$, there exists an idempotent $(\mu, m-1, m)$-latin trade whenever $1 \leq \mu \leq m-2$.*

Proof. It was shown in [80] that for $m \neq 6, 14, 62$ there exists a large set of idempotent latin squares of order m. The cases $m = 14, 62$ were solved in [32] and [31] respectively. By taking such a large set and deleting the cells of the main diagonals of each of the idempotent latin squares, we have an idempotent $(m-2, m-1, m)$-latin trade for $m \geq 3$, $m \neq 6$. Clearly we can remove any number of the resulting partial latin squares to yield an idempotent $(\mu, m-1, m)$-latin trade for $1 \leq \mu \leq m-2$. $\square$

Generalizing from the method of finding pairs of transversals of given intersection in the back-circulant latin squares, in chapter 2 we were able to determine the possible intersection sizes of three transversals in the back circulant latin square. We restate the result:

Theorem 3.3.6. *For odd integer $m \geq 33$ with $m = 18I + 9 + 2d$, $0 \leq d < 9$, and $m = 22I' + 11 + 2d'$, $0 \leq d' < 11$, there exists three transversals of the back circulant latin square B_m, T_1, T_2, T_3, for each $t \in \{\min(d'+3, d), \dots, m\} \setminus \{m-5, \dots, m-1\}$ such that $S = T_1 \cap T_2 = T_1 \cap T_3 = T_2 \cap T_3$ and $|S| = t$, except possibly when:*

- *$m = 51$ and $t = 29$,*

- $m = 53$ *and* $t = 30$.

Further, a transformation was provided to construct (μ, k, m)-latin trades from a collection of μ transversals of the back circulant latin square:

Theorem 3.3.7. *Take m odd and $0 \leq t \leq m$. If there exists a set $S \subseteq [m]^3$ with $|S| = t$ and μ transversals of B_m, $T_1, \ldots, T_\mu$ with $T_\alpha \cap T_\beta = S$, for $\alpha, \beta \in [\mu]$ and $\alpha \neq \beta$, then there exists a circulant $(\mu, m-t, m)$-latin trade.*

Proof. Consider the μ partial latin squares defined by $Q_\alpha = \{(i, c+i, r+c+i) \mid i \in [m], (r, c, r+c) \in T_\alpha \setminus S\}$. The set $(Q_1, \ldots, Q_\mu)$ forms a $(\mu, m-t, m)$-latin trade that is circulant by definition. □

This can be generalized for our purposes in the following manner:

Theorem 3.3.8. *Take m odd and $1 \leq t \leq m$. If there is a set $S \subseteq [m]^3$ with $|S| = t$ such that there exists μ transversals of B_m, $T_1, \ldots, T_\mu$, with $T_\alpha \cap T_\beta = S$, for $\alpha, \beta \in [m]$ and $\alpha \neq \beta$, then there exists an idempotent circulant $(\mu, m-t, m)$-latin trade.*

Proof. Notice that $T_\alpha^{x,y} = \{(r+x, c+y, r+c+x+y) \mid (r, c, r+c) \in T_\alpha\}$, for $\alpha \in [\mu]$ and $x, y \in \{0, \ldots, m-1\}$, will define a new collection of μ transversals of B_m with $T_\alpha^{x,y} \cap T_\beta^{x,y} = S^{x,y}$ such that $|S^{x,y}| = t$, for $\alpha, \beta \in [\mu]$ and $\alpha \neq \beta$. As $t \geq 1$, this allows us to assume without loss of generality that $(m, m, m) \in T_\alpha$, for all $\alpha \in [\mu]$, or equivalently $(m, m, m) \in S$. Applying the construction from Theorem 3.3.7 to these transversals of B_m, we obtain a circulant $(\mu, m-t, m)$-latin trade, which we denote as $\mathcal{Q} = (Q_1, \ldots, Q_\mu)$. As $(m, m, m) \in S$, then each $(r, c, r+c) \in T_\alpha \setminus S$ has $r+1 \not\equiv 1$, $c+1 \not\equiv 1$, and $r+c+1 \not\equiv 1$. Then $\mathcal{Q}$ has its $(1,1)$ cell empty as $c+1 \not\equiv 1$, and the symbol 1 will not appear in the first row as $r+c+1 \not\equiv 1$. The first column of Q_α contains cells $\{(m-c+1, 1, r+1) \mid (r, c, r+c) \in T_\alpha \setminus S\}$, for $\alpha \in [\mu]$. As $r+1 \not\equiv 1$, the symbol 1 does not appear in the first column. By the circulant nature, we also have $(i,i) \notin \mathcal{S}(\mathcal{Q})$ and $i \notin \mathcal{R}_i(\mathcal{Q}) \cap \mathcal{C}_i(\mathcal{Q})$, for $i \in [m]$. Then $\mathcal{Q}$ is an idempotent circulant $(\mu, m-t, m)$-latin trade. $\square$

We can then exploit the existence of two [30] and three (see chapter 2) transversals of given intersection as:

Theorem 3.3.9. *For odd integer $m \geq 5$, there exists an idempotent circulant $(2, m-t, m)$-latin trade for $t \in [m] \setminus \{m-2, m-1\}$ except, perhaps, when $(t,m) = (2,5)$.*

Theorem 3.3.10. *For odd integer $m \geq 33$ with $m = 18I + 9 + 2d$, $0 \leq d < 9$, and $m = 22I' + 11 + 2d'$, $0 \leq d' < 11$, there exists an idempotent circulant $(3, m-t, m)$-latin trade, for $t \in \{\max(1, \min(d'+3, d)), \ldots, m\} \setminus \{m-5, \ldots, m-1\}$, except, perhaps, when:*

- *$m = 51$ and $t = 29$,*
- *$m = 53$ and $t = 30$.*

3.4 New Constructions for idempotent (μ, k, m)-latin trades

In this section, we will consider block theoretic constructions that are able to determine the spectrum of $(3, k, m)$-latin trades for all but a small list of values of k and m.

3.4.1 Computer search for small orders

If $B = \{(a_1, \ldots, a_\mu)_{c_l} \mid 1 \leq l \leq k\}$, Algorithm 1 of [7] can be used to show B is the base row of a (μ, k, m)-latin trade. If for each $(a_1, \ldots, a_\mu)_{c_l}$ it further holds that:

- $a_\alpha \neq 1$, for all $\alpha \in [\mu]$;
- $c_l \neq 1$; and
- $a_\alpha \neq c_l$,

then B is the base row of an idempotent (μ, k, m)-latin trade. As the result of a computational search, we introduce the following base rows of idempotent $(3, k, m)$-latin trades:

- $k = 5$

$$
\begin{aligned}
3 - IB_7^5 =& \{(3,4,5)_2, (5,7,4)_3, (7,5,2)_4, (2,3,7)_5, (4,2,3)_6\} \\
3 - IB_8^5 =& \{(3,4,6)_2, (8,2,4)_3, (6,8,3)_4, (4,6,2)_5, (2,3,8)_6\} \\
3 - IB_9^5 =& \{(3,6,9)_2, (6,2,7)_3, (2,7,3)_4, (9,3,6)_5, (7,9,2)_8\} \\
3 - IB_{11}^5 =& \{(5,7,9)_2, (7,2,8)_3, (9,8,7)_4, (2,9,5)_6, (8,5,2)_9\} \\
3 - IB_{12}^5 =& \{(4,5,8)_2, (11,2,5)_3, (8,11,4)_5, (5,8,2)_6, (2,4,11)_8\}
\end{aligned}
$$

- $k = 6$

$$
\begin{aligned}
3 - IB_8^6 =& \{(3,4,6)_2, (5,8,2)_3, (8,2,5)_4, (2,6,3)_5, (4,5,8)_6, (6,3,4)_7\} \\
3 - IB_9^6 =& \{(3,6,7)_2, (7,4,5)_3, (6,2,3)_4, (4,7,6)_5, (2,5,4)_6, (5,3,2)_7\} \\
3 - IB_{10}^6 =& \{(5,8,9)_2, (9,2,4)_3, (2,5,3)_4, (4,3,8)_5, (3,9,2)_6, (8,4,5)_7\} \\
3 - IB_{11}^6 =& \{(3,4,10)_2, (6,9,4)_3, (10,2,6)_4, (2,6,3)_5, (4,3,9)_6, (9,10,2)_7\} \\
3 - IB_{12}^6 =& \{(3,9,11)_2, (7,2,5)_3, (11,5,3)_4, (2,7,9)_5, (5,3,7)_6, (9,11,2)_7\} \\
3 - IB_{13}^6 =& \{(8,11,13)_2, (13,8,9)_3, (9,2,8)_4, (3,9,2)_5, (2,3,11)_6, (11,13,3)_7\}
\end{aligned}
$$

- $k = 7$

$$
\begin{aligned}
3 - IB_m^7 =& \{(3,4,6)_2, (7,6,5)_3, (6,2,7)_4, (2,9,3)_5, (9,7,2)_6, (5,3,4)_7, (4,5,9)_8\}, \\
& \text{for } m \geq 9
\end{aligned}
$$

- $k = 8$

$$3 - IB_m^8 = \{(3,4,5)_2, (2,6,7)_3, (7,8,2)_4, (9,2,6)_5, (8,5,3)_6, (4,3,9)_7, (6,9,4)_8, (5,7,8)_9\}, \text{ for } m \geq 10$$

- $k = 9$

$$3 - IB_m^9 = \{(3,4,5)_2, (5,8,7)_3, (7,2,9)_4, (9,6,2)_5, (11,9,8)_6, (2,11,3)_7, (4,3,6)_8, (6,5,4)_9, (8,7,11)_{10}\}, \text{ for } m \geq 11$$

- $k = 10$

$$3 - IB_m^{10} = \{(3,4,5)_2, (2,6,4)_3, (6,5,9)_4, (9,10,2)_5, (11,2,10)_6, (10,11,3)_7, (4,3,7)_8, (7,8,11)_9, (5,7,8)_{10}, (8,9,6)_{11}\}, \text{ for } m \geq 12$$

- $k = 11$

$$3 - IB_m^{11} = \{(3,4,5)_2, (5,6,8)_3, (7,2,10)_4, (9,11,3)_5, (11,10,2)_6, (13,8,11)_7, (2,13,9)_8, (4,3,6)_9, (6,5,4)_{10}, (8,7,13)_{11}, (10,9,7)_{12}\}, \text{ for } m \geq 13$$

- $k = 12$

$$3 - IB_m^{12} = \{(3,4,5)_2, (2,6,4)_3, (6,2,3)_4, (8,9,11)_5, (11,12,8)_6, (13,3,12)_7, (12,13,2)_8, (4,10,13)_9, (7,5,6)_{10}, (5,8,9)_{11}, (10,11,7)_{12}, (9,7,10)_{13}\}, \text{ for } m \geq 14$$

- $k = 13$

$$3 - IB_m^{13} = \{(3,4,5)_2, (5,6,4)_3, (2,5,10)_4, (9,10,12)_5, (11,13,2)_6, (13,2,9)_7, (15,12,13)_8, (12,15,3)_9, (4,3,8)_{10}, (6,7,15)_{11}, (8,9,7)_{12}, (10,11,6)_{13}, (7,8,11)_{14}\}, \text{ for } m \geq 15$$

- $k = 14$

$$3 - IB_m^{14} = \{(3,4,5)_2, (2,6,4)_3, (6,2,3)_4, (8,9,2)_5, (10,3,13)_6, (13,14,12)_7, (15,13,14)_8, (14,15,11)_9, (4,5,6)_{10}, (7,12,15)_{11}, (5,11,10)_{12}, (11,7,8)_{13}, (9,10,7)_{14}, (12,8,9)_{15}\}, \text{ for } m \geq 16$$

- $k = 15$

$$3 - IB_m^{15} = \{(3,4,5)_2, (5,6,4)_3, (2,5,6)_4, (8,2,9)_5, (11,13,14)_6, (13,15,12)_7, (15,12,2)_8, (17,14,15)_9, (14,3,17)_{10}, (4,17,3)_{11}, (6,7,10)_{12}, (10,11,8)_{13}, (9,10,7)_{14}, (7,9,11)_{15}, (12,8,13)_{16}\}, \text{ for } m \geq 17$$

- $k = 16$

$$3 - IB_m^{16} = \{(3,4,5)_2, (2,6,4)_3, (6,2,3)_4, (8,9,2)_5, (4,3,10)_6, (13,12,15)_7, (15,16,13)_8, (17,15,16)_9, (14,17,6)_{10}, (16,5,17)_{11}, (5,7,14)_{12}, (7,14,8)_{13}, (10,13,12)_{14}, (12,11,7)_{15}, (11,8,9)_{16}, (9,10,11)_{17}\}, \text{ for } m \geq 18$$

- $k = 17$

$$3 - IB_{19}^{17} = \{(3,4,5)_2, (5,6,4)_3, (2,5,6)_4, (8,2,3)_5, (10,11,12)_6, (13,14,15)_7, (15,17,19)_8, (17,13,14)_9, (19,16,17)_{10}, (16,19,2)_{11}, (4,3,16)_{12}, (6,8,7)_{13}, (9,7,11)_{14}, (11,9,10)_{15}, (7,12,9)_{16}, (14,15,13)_{17}, (12,10,8)_{18}\}$$

$$3 - IB_m^{17} = \{(3,4,5)_2, (2,6,4)_3, (6,2,3)_4, (8,9,7)_5, (10,5,11)_6, (14,15,16)_7, (16,17,15)_8, (18,14,17)_9, (15,16,14)_{10}, (17,18,2)_{11}, (4,13,18)_{12}, (7,3,8)_{13}, (12,8,10)_{14}, (5,11,13)_{15}, (11,7,6)_{16}, (13,12,9)_{17}, (9,10,12)_{18}\}, \text{ for } m \geq 20$$

- $k = 18$

$$\begin{aligned}3 - IB_m^{18} = &\{(3,4,5)_2, (2,6,4)_3, (6,2,3)_4, (8,9,2)_5, (4,3,8)_6, (11,8,12)_7,\\ &(15,16,17)_8, (17,18,16)_9, (19,15,18)_{10}, (16,17,15)_{11}, (18,19,6)_{12},\\ &(5,7,19)_{13}, (7,5,9)_{14}, (9,14,13)_{15}, (12,11,7)_{16}, (14,13,10)_{17},\\ &(13,10,14)_{18}, (10,12,11)_{19}\}, \text{ for } m \geq 20\end{aligned}$$

Theorem 3.4.1. *There exist idempotent* $(3, k, m)$*-latin trades for* $m \geq k + 1$ *when:*

- $k = 4$ *and* $5|m$*;*
- $k = 5$*, except when* $m = 6$ *and perhaps when* $m = 10, 13$*; and*
- $6 \leq k \leq 18$*.*

Proof. The previously stated base rows, along with the idempotent $(3, m-1, m)$-latin trades of Theorem 3.3.5 complete the cases for $7 \leq k \leq 18$. For $k = 6$, we can use the previously stated base rows along with an idempotent $(3, 6, 7)$-latin trade from Theorem 3.3.5 with Theorem 3.3.4. For $k = 5$, there exists idempotent $(3, 5, m)$-latin trades for $m \in \{7, 8, 9, 11, 12\}$. Using these $(3, 5, m)$-latin trades in Theorem 3.3.1 will, after two iterations, yield the required $(3, 5, m)$-latin trades for Theorem 3.3.4 with $k' = 14$. For $k = 4$, there exists an idempotent $(3, 4, 5)$-latin trade by Theorem 3.3.5, on which we can repeatedly apply Theorem 3.3.1 to get the result. If an idempotent $(3, 5, 6)$-latin trade existed, then adding the cells $\{(i, i, i) \mid 1 \leq i \leq n\}$ to each of the partial latin squares of the idempotent $(3, 5, 6)$-latin trade would yield an ordered triple of idempotent latin squares of order 6 that are pairwise disjoint in each cell not on the main diagonal. Using a computer we searched for such ordered triples of idempotent latin squares by testing all possible triples of idempotent latin squares of order 6. None existed, and so there does not exists an idempotent $(3, 5, 6)$-latin trade. □

We conjecture that the two unresolved cases with $k = 5$ and $m = 10, 13$ both exist.

3.4.2 Extended Multiplication Construction

Lemma 3.4.2. *Take $n \geq 3$ and $m \geq 4$. If $n = 6$, let y be a positive integer with $\mu < y \leq \frac{m}{4}$. If $n \neq 6$, let y be a positive integer with $\mu < y \leq \frac{m}{2}$. If there exists an idempotent (μ, k, m)-latin trade for each $k \in \{0, y, y+1, \ldots, m-1\}$, then there exists an idempotent (μ, k, mn)-latin trade for each $k \in \{0, y, y+1, \ldots, mn-1\}$.*

Proof. In the case that $n \neq 6$, Theorem 3.3.3 yields idempotent (μ, k, mn)-latin trades for $k \in \{\sum_{i=1}^{n} y_i \mid y_1 \in \{0, y, y+1, \ldots, m-1\}$ and $y_i \in \{0, y, y+1, \ldots, m\}, 2 \leq i \leq n\} = \{0, y, y+1, \ldots, mn-1\}$ (y_i, with $2 \leq i \leq n$, may equal m as there exists a $(3, m, m)$-latin trade by Theorem 3.2.6).

Now we consider the case when $n = 6$. Applying Theorem 3.3.2 using an idempotent (μ, k, m)-latin trade and a $(1, 2, 2)$-latin trade (which exists by Theorem 3.2.5) yields an idempotent $(\mu, 2k, 2m)$-latin trade for each k with $y \leq k \leq m-1$. Applying Theorem 3.3.1 to the idempotent (μ, k, m)-latin trades yields idempotent $(\mu, k, 2m)$-latin trades for each k with $y \leq k \leq m-1$. Considering these, along with the existence of an idempotent $(\mu, 2m-1, 2m)$-latin trade from Theorem 3.3.5, yields idempotent $(\mu, k', 2m)$-latin trades for $k' \in \Gamma = \Gamma_1 \cup \Gamma_2$, where $\Gamma_1 = \{0\} \cup \{y, y+1, \ldots, m-1\}$ and $\Gamma_2 = \{2y, 2y+2, \ldots, 2m-2\} \cup \{2m-1\}$. Applying Theorem 3.3.3 with $l = 3$, using the above idempotent $(\mu, k', 2m)$-latin trades with $k' \in \Gamma$ along with $(\mu, 2m, 2m)$-latin trades and $(\mu, m, 2m)$-latin trades that exist by Theorem 3.2.6 and Theorem 3.3.1, yields idempotent $(\mu, k'', 6m)$-latin trades for each $k'' \in \{\sum_{i=1}^{3} y_i \mid y_1 \in \Gamma$ and $y_2, y_3 \in \Gamma \cup \{m, 2m\}\}$.

It holds that $\{\sum_{i=1}^{3} y_i \mid y_1 \in \Gamma_2$ and $y_2, y_3 \in \Gamma_2 \cup \{2m\}\} = \{6y, 6y+2, \ldots, 4y+2m-4, 4y+2m-2\} \cup \{4y+2m-1, 4y+2m, \ldots, 6m-1\}$, and also that $\{\sum_{i=1}^{3} y_i \mid y_1 \in \Gamma_1$ and $y_2, y_3 \in \Gamma_1 \cup \{m\}\} = \{0\} \cup \{y, y+1, \ldots, m\} \cup \{2y, 2y+1, \ldots, 2m\} \cup \{3y, 3y+1, \ldots, 3m-1\} = \{0\} \cup \{y, y+1, \ldots, 3m-1\}$ as $y \leq m/4$ implies both $m \geq 2y$ and $2m \geq 3y$. Then it holds that $\{\sum_{i=1}^{3} y_i \mid y_1 \in \Gamma$ and $y_2, y_3 \in \Gamma \cup \{m, 2m\}\} \supseteq \{0\} \cup \{y, y+1, \ldots, 3m-1\} \cup \{4y+$

$2m-1, \ldots, 6m-1\} = \{0\} \cup \{y, \ldots, 6m-1\}$, as $m \geq 4y$, and so the proof is complete. $\square$

There does not exist a pair of orthogonal latin squares of order 2, so we do not have a similar result to this lemma when $n = 2$. This leaves us with two cases that are of particular interest, as they are not covered by Lemma 3.4.2: $m = p$ and $m = 2p$, for p a prime. The next two subsection contain constructions that will be used to fill the spectrum $\mathcal{IS}^3_m$ for certain $m = p$ and $m = 2p$.

3.4.3 Packing construction

The following theorem uses μ-way latin trades of order λ and volume s in the construction of (μ, s, m)-latin trades, for certain integers $m \geq \lambda^2 + 2\lambda + 1$. Afterwards we will modify the resulting structures to yield idempotent (μ, s, m)-latin trades. The 3-way intersection problem for latin squares has been studied previously, and this will yield the 3-way latin trades we need in order to apply this construction, which we detail later.

Theorem 3.4.3. *Suppose there exists a μ-way latin trade of volume s and of order λ. For every $m = \lambda(\lambda + a) + b$, where $0 < b < \lambda$, $a \geq b + 1$, and $\gcd(m, \lambda) = \gcd(\lambda, b) = 1$, there exists a (μ, s, m)-latin trade.*

In order to prove this theorem, we construct the (μ, s, m)-latin trade as follows.

Construction 3.4.4. *Suppose there exists a μ-way latin trade $\mathcal{U} = (U_1, \ldots, U_\mu)$ of volume s, order λ, and using symbol set $\Omega = [\lambda]$. Let $m = \lambda(\lambda + a) + b$, where a and b are integers with $0 < b < \lambda$, $a \geq b + 1$, and $\gcd(m, \lambda) = \gcd(\lambda, b) = 1$. Let $U_\alpha[f]$ be the array obtained from U_α by replacing each occurrence of symbol i with symbol $b_i + f \pmod{m}$, where $b_i = i(\lambda + a - 1)$ for each $1 \leq i \leq \lambda$, $\alpha \in [\mu]$, and $0 \leq f < m$.*

We construct R_α, an $m \times m$ array of cells for each $\alpha \in [\mu]$. For each $f \in \{0, \ldots, m-1\}$, consider the $\lambda \times \lambda$ block of cells within R_α given by $B_f = \{(i, j) \mid \lambda f < i \leq \lambda(f + 1), f <$

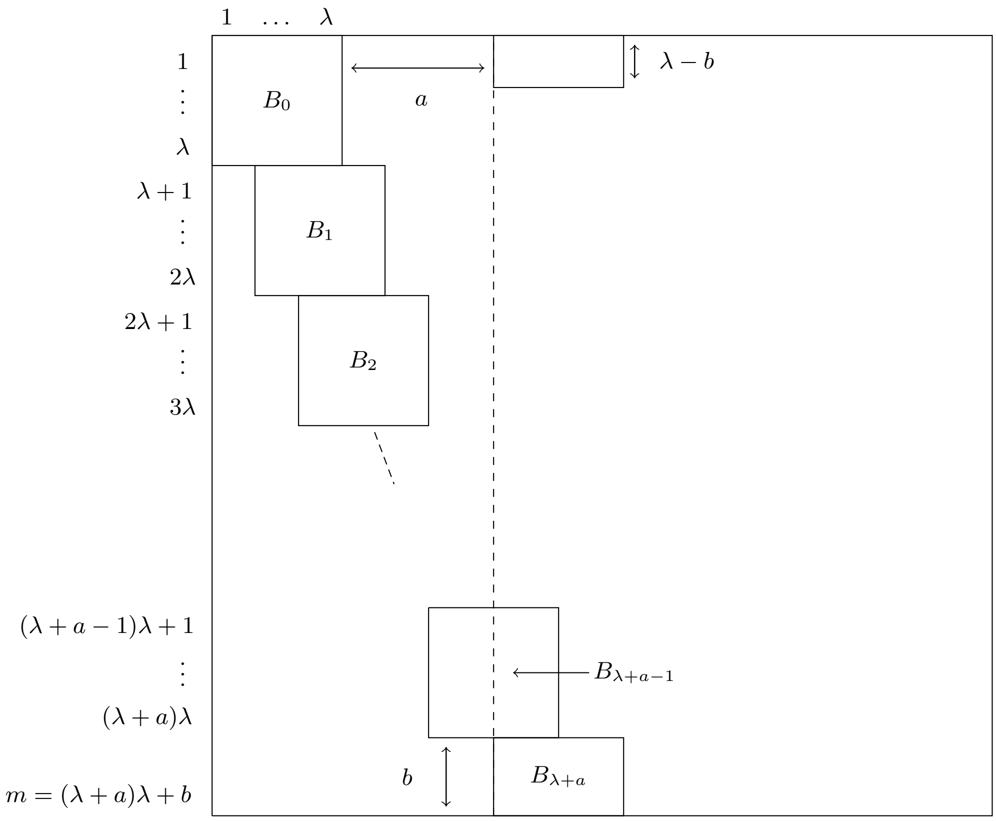

Figure 3.1: An illustrative example of the placement of blocks in Construction 3.4.4

$j \leq f+\lambda\}$, with i,j taken mod m, but 0 is identified with m such that $1 \leq i,j \leq m$. This placement of the blocks of cells in R_α is displayed visually in Figure 3.1. Notice $B_f \cap B_g = \emptyset$ for each $0 \leq f,g < m$, $f \neq g$. Place $U_\alpha[f]$ into the cells B_f of R_α, for each $0 \leq f < m$, and leave every other cell empty. Then cell $(\lambda f+i', f+j') \in B_f$ is filled in R_α with the symbol $U_\alpha[f](i',j')$, for $1 \leq i',j' \leq \lambda$.

The following proof verifies that the R_α form a (μ,s,m)-latin trade.

Proof. We will show that the collection of μ $m \times m$ arrays $\mathcal{R} = (R_1,\dots,R_\mu)$ defined by Construction 3.4.4 form a (μ,s,m)-latin trade. We begin by showing that $\mathcal{R}$ forms a

μ-way latin trade of order m. This amounts to showing that each R_α forms a partial latin square, as then by construction it is clear that the μ partial latin squares form a μ-way latin trade of order m. To this end, we must verify that any symbol appears in a column of R_α at most once, and that any symbol appears in a row of R_α at most once.

To show the symbols that appear in a column of R_α are distinct, we will consider a specific symbol $b_\lambda = \lambda(\lambda + a - 1)$. This symbol appears only in the block of cells $B_{j(\lambda+a-1)}$ of R_α, for $0 \leq j \leq \lambda - 1$. The columns of the block of cells $B_{j(\lambda+a-1)}$ of R_α are exactly those columns c with $j(\lambda + a - 1) < c \leq j(\lambda + a - 1) + \lambda$. For $0 \leq j \leq \lambda - 1$, the sets of integers $\{j(\lambda + a - 1) + 1, \ldots, j(\lambda + a - 1) + \lambda\}$ are each disjoint, and in the range 1 to m. That is to say that two distinct block of cells B_f and $B_{f'}$ containing b_λ do not intersect column-wise, and as a column within each block of cells B_f of R_α can contain b_λ at most once, we can conclude that b_λ appears in each column of R_α at most once. By construction, $(r, c, b_\lambda) \in R_\alpha$ if and only if $(r + \lambda, c + 1, b_\lambda + 1) \in R_\alpha$, and so symbol $b_\lambda + 1$ appears in each column of R_α at most once. Repeating this argument, we see that every symbol will appear in each column of R_α at most once.

To show the symbols that appear in a row of R_α are distinct, we consider a specific symbol $a - b - 1 = b_\lambda + (\lambda + a - 1) - m$, which we denote as $b_{\lambda+1}$. This symbol appears in the block of cells $B_{j(\lambda+a-1)}$ of R_α, for $1 \leq j \leq \lambda$. The rows of the subsquare $B_{j(\lambda+a-1)}$ of R_α are exactly those rows r with $j\lambda(\lambda + a - 1) < r \leq j\lambda(\lambda + a - 1) + \lambda$, or equivalently $m - j(\lambda + b) < r \leq m - j(\lambda + b) + \lambda$ once we consider r to be taken modulo m. For $1 \leq j \leq \lambda$, the sets of integers $\{m - j(\lambda + b) + 1, \ldots, m - j(\lambda + b) + \lambda\}$ are each disjoint. As $\lambda(\lambda + b) < \lambda(\lambda + a) < m$, these sets of integers only contain values in the range 1 to m. That is to say that two distinct blocks of cells B_f and $B_{f'}$ that both contain $b_{\lambda+1}$ do not intersect row-wise, so we can conclude that $b_{\lambda+1}$ appears in each row of R_α at most once. By construction, $(r, c, b_{\lambda+1}) \in R_\alpha$ if and only if $(r + \lambda, c + 1, b_{\lambda+1} + 1) \in R_\alpha$, and so $b_{\lambda+1} + 1$ appears in each row of R_α at most once. Repeating this argument, we see that every symbol will appear in each row of R_α at most once. We have now shown that the R_α form a μ-way latin trade of order m.

Now we show that R_α is s-homogeneous, for each $\alpha \in [\mu]$. For $\alpha \in [\mu]$, the construction filled each of the m blocks B_f of R_α with s cells, for $0 \leq f < m$. As no overlap occurs between the blocks B_f, R_α was filled by precisely sm filled cells. By construction, if cell (r, c) is filled in R_α, then it holds that $(r, c, e) \in R_\alpha$ if and only if $(r+\lambda, c+1, e+1) \in R_\alpha$. Then row r (column c, symbol e) must contain the same number of filled cells as row $r+\lambda$ (column $c+1$, symbol $e+1$). We can repeat this argument $m-1$ times to show each row and column will have the same number of filled cells, and that each each symbol will have the same number of occurrences in R_α (for the row case, we have used the assumption that $\gcd(m, \lambda) = 1$). Then this implies the sm filled cells are spread evenly amongst the m rows, columns, and symbols. This gives s filled cells per row, s filled cells per column, and s occurrences per symbol.

This shows that each R_α is s-homogeneous, and so the proof is complete. □

Example 3.4.5. *We demonstrate this technique using a 2-way latin trade of volume $s = 7$ and of order $\lambda = 3$ given by the pair of partial latin squares:*

1	2	3
3		1
2	3	

2	3	1
1		3
3	2	

We will take $a = 2$ and $b = 1$, giving the order of the resulting trade as $\lambda(\lambda + a) + b = 16$. Then $b_1 = 4$, $b_2 = 8$, and $b_3 = 12$. Using the first of the above partial latin squares of order 3 and volume 7 in the construction gives a partial latin square of order 16 that is 7-homogeneous:

4	*8*	*12*			*1*		*9*			*2*	*6*				
12		*4*			*13*	*1*					*15*	*3*	*7*		
8	*12*					*10*	*14*	*2*			*7*		*15*		
	5	*9*	*13*			*2*		*10*			*3*	*7*			
	13		*5*			*14*	*2*					*16*	*4*	*8*	
	9	*13*					*11*	*15*	*3*			*8*		*16*	
		6	*10*	*14*			*3*		*11*			*4*	*8*		
		14		*6*			*15*	*3*					*1*	*5*	*9*
		10	*14*					*12*	*16*	*4*			*9*		*1*
			7	*11*	*15*			*4*		*12*			*5*	*9*	
10			*15*		*7*			*16*	*4*					*2*	*6*
2			*11*	*15*					*13*	*1*	*5*			*10*	
				8	*12*	*16*			*5*		*13*			*6*	*10*
7	*11*			*16*		*8*			*1*	*5*					*3*
	3			*12*	*16*					*14*	*2*	*6*			*11*
11					*9*	*13*	*1*			*6*		*14*			*7*

We can use the second partial latin square of order 3 *and volume* 7 *to construct a similar partial latin square of order* 16 *that is* 7*-homogeneous. Together, these partial latin squares form a* 2*-way* 7*-homogeneous latin trade of order* 16*.*

Theorem 3.4.6. *Suppose there exists a μ-way latin trade of volume s and of order λ, with $\lambda \geq 1$. For every $m = \lambda(\lambda + a) + b$, where $0 < b < \lambda$, $a \geq b + 1$, and $\gcd(m, \lambda) = \gcd(\lambda, b) = 1$, there exists a circulant idempotent (μ, s, m)-latin trade.*

Proof. Consider the μ arrays $R_1, \ldots, R_\mu$ from Construction 3.4.4. Define the array $\bar{R}_\alpha$ by the set of ordered triples $\bar{R}_\alpha = \{(\sigma_1(r), c, \sigma_2(e)) \mid (r, c, e) \in R_\alpha\}$ with $\sigma_1(r) = \lambda^{-1} \cdot (r-1) - 1 \pmod{m}$ and $\sigma_2(e) = e - 2(\lambda + a - 1) \pmod{m}$, for each $\alpha \in [\mu]$, where λ^{-1} is the unique inverse of $\lambda \pmod{m}$ which exists by the assumption that $\gcd(m, \lambda) = 1$. Both

σ_1 and σ_2 are permutations of $[m]$. As the R_α form a (μ, s, m)-latin trade, the $\bar{R}_\alpha$ also form a (μ, s, m)-latin trade as the three properties of Definition 3.1.1 are invariant under permutation swaps of the rows, columns, and symbols.

Since $(r, c, e) \in R_\alpha$ implies $(r + \lambda, c + 1, e + 1) \in R_\alpha$ by construction, it follows that $(r, c, e) \in \bar{R}_\alpha$ implies $(r + 1, c + 1, e + 1) \in \bar{R}_\alpha$, and so $\bar{R}_\alpha$ is a circulant (μ, s, m)-latin trade.

We show that $1 \notin \mathcal{R}_1(\bar{R}_\alpha)$, $1 \notin \mathcal{C}_1(\bar{R}_\alpha)$, and $(1, 1) \notin \mathcal{S}(\bar{R}_\alpha)$. Noting that $\sigma_1^{-1}(1) = 2\lambda + 1$ and $\sigma_2^{-1}(1) = 2\lambda + 2a - 1$, this is equivalent to showing $2\lambda + 2a - 1 \notin \mathcal{R}_{2\lambda+1}(R_\alpha)$, $2\lambda + 2a - 1 \notin \mathcal{C}_1(R_\alpha)$, and $(2\lambda + 1, 1) \notin \mathcal{S}(R_\alpha)$.

We first show that $2\lambda + 2a - 1 \notin \mathcal{R}_{2\lambda+1}(R_\alpha)$. The symbol $2\lambda + 2a - 1$ appears only in the blocks $B_{j(\lambda+a-1)+b+3\lambda+2a-1}$ of R_α for $0 \le j \le \lambda - 1$, hence it only appears within the rows $T = \cup_{j=0}^{\lambda-1} \{\lambda(j(\lambda + a - 1) + b + 3\lambda + 2a - 1) + 1, \ldots, \lambda(j(\lambda + a - 1) + b + 3\lambda + 2a - 1) + \lambda\}$. If we perform a change in variables, sending j to $\lambda - 2 - j$, then $T = \cup_{j=0}^{\lambda-2} \{\lambda + 1 + j(\lambda + b), \ldots, 2\lambda + j(\lambda + b)\} \cup \{m - b + 1, \ldots, m\} \cup \{1, \ldots, \lambda - b\}$, which does not contain $2\lambda + 1$. Then the symbol $2\lambda + 2a - 1$ does not appear in the row $2\lambda + 1$.

Secondly we show that $2\lambda + 2a - 1 \notin \mathcal{C}_1(R_\alpha)$. The symbol $2\lambda + 2a - 1$ appears exactly in the blocks $B_{j(\lambda+a-1)+b+3\lambda+2a-1}$ of R_α, for $0 \le j \le \lambda - 1$. These blocks only use the columns $\cup_{j=0}^{\lambda-1} \{j(\lambda + a - 1) + b + 3\lambda + 2a, \ldots, j(\lambda + a - 1) + b + 3\lambda + 2a + \lambda - 1\} = \cup_{j=0}^{\lambda-3} \{j(\lambda + a - 1) + b + 3\lambda + 2a, \ldots, j(\lambda + a - 1) + b + 3\lambda + 2a + \lambda - 1\} \cup \{2, \ldots, \lambda + 1\} \cup \{\lambda + a + 1, \ldots, \lambda + a + \lambda\}$. As such, the column with index 1 does not contain $2\lambda + 2a - 1$.

Thirdly we show that $(2\lambda + 1, 1) \notin \mathcal{S}(R_\alpha)$. Suppose for the sake of contradiction that $(2\lambda + 1, 1) \in \mathcal{S}(R_\alpha)$. Then there must be some block $B_{f'}$ that contains cell $(2\lambda + 1, 1)$, $0 \le f' \le m - 1$. As $B_1 = \{(i, j) \mid \lambda + 1 \le i \le 2\lambda, 2 \le j \le 1 + \lambda\}$, $B_{f'}$ cannot contain the cell $(2\lambda, 2)$. Then $B_{f'}$ must either have exactly the rows $\{2\lambda + 1, \ldots, 3\lambda\}$, or have exactly the columns $\{m - \lambda + 2, \ldots, m\} \cup \{1\}$.

The former implies $B_{f'}$ contains exactly the same rows as B_2. As the first rows are the same, $f'\lambda+1 \equiv 2\lambda+1 \mod m$, and as $\gcd(\lambda, m) = 1$, we have $f' = 2$. But then B_2 must contain cell $(2\lambda + 1, 1)$, which when we look at the columns implies $\lambda + 2 \geq m + 1$. As $m \geq \lambda(\lambda + 2) + 1$ and $\lambda \geq 1$, this is impossible.

The later implies $B_{f'} = B_{1-\lambda} = B_{m+1-\lambda}$. Then each of the m rows are represented at least once in the set of $\lambda + 1$ blocks $\{B_{m+1-\lambda}, B_{m+2-\lambda}, \dots, B_{m-1}\} \cup \{B_0, B_1\}$, but these $\lambda + 1$ blocks only use λ rows each, and so $\lambda(\lambda + 1)$ rows in total. This implies $m \leq \lambda(\lambda + 1)$, which contradicts the fact that $m = \lambda(\lambda + a) + b$ and that $a > b \geq 1$.

Then none of the cases are possible, forming a contradiction, and so $(2\lambda + 1, 1) \notin \mathcal{S}(R_\alpha)$. This completes the proof. □

The three-way intersection problem for latin squares has been studied in [1], where the authors consider three latin squares L_1, L_2, L_3 of order n with common intersection $P = L_1 \cap L_2 = L_1 \cap L_3 = L_2 \cap L_3$. The collection of partial latin squares $(L_1 \setminus P, L_2 \setminus P, L_3 \setminus P)$ forms a 3-way latin trade of volume $n^2 - |P|$ and of any order n' greater than or equal to n (as we allow empty rows and columns in 3-way latin trades). We can thus interpret the results of [1] in terms of 3-way latin trades and combine them with Theorem 3.4.6 to yield:

Theorem 3.4.7. *For $\lambda \geq 3$, there exists circulant idempotent $(3, k, m)$-latin trades for $m = \lambda(\lambda + a) + b$, where $0 < b < \lambda$, $\gcd(\lambda, b) = 1$, and $a \geq b + 1$, and:*

- *$k \in \{0, 9\}$, for $\lambda = 3$;*
- *$k \in \{0, 9, 12, 15, 16\}$, for $\lambda = 4$;*
- *$k \in \{0, 9, 12, 15, 16\}$ or $18 \leq k \leq 25$, for $\lambda = 5$;*
- *$k \in \{0, 9, 12\}$ or $15 \leq k \leq \lambda^2$, for $\lambda \geq 6$.*

3.4.4 Construction via RPBDs

Definition 3.4.8. *A* (v, M, λ) pairwise balanced design*, denoted* $\mathrm{PBD}(v, M, \lambda)$*, is a pair* (V, B)*, with* V *a set of* v *symbols and* B *a set of subsets of* V *(each subset is called a block) with sizes from* M*, such that each pair of elements of* V *can be found in exactly* λ *blocks of* B*.*

Definition 3.4.9. *A* resolvable (v, M, λ, n) pairwise balanced design*, which we denote by* $\mathrm{RPBD}(v, M, \lambda, n)$*, is a pair* (V, B) *along with* n *resolution classes* $R_1, \dots, R_n$*, such that* (V, B) *is a* $\mathrm{PBD}(v, M, \lambda)$*, the sets* $R_1, \dots, R_n$ *partition* B*, and each symbol appears in precisely one block of each resolution class.*

Theorem 3.4.10. *Suppose there exists a* $\mathrm{RPBD}(v, M, 1, n+1)$*,* (V, B)*, with resolution classes* $R_1, \dots, R_n$ *and* R_∞*. Suppose there exists integers* $d_i \geq 1$*,* $1 \leq i \leq n$*, such that for each* $b \in R_i$ *there exists an idempotent* $(\mu, |b| - d_i, |b|)$*-latin trade and integer* $d_\infty \geq 0$ *such that for each* $b \in R_\infty$ *there exists a* $(\mu, |b| - d_\infty, |b|)$*-latin trade. Then there exists a* $(\mu, v + n - \sum_{i=1}^{n} d_i - d_\infty, v)$*-latin trade.*

The following construction was suggested by Prof. L. Zhu, which is a modification of a construction for sets of idempotent latin squares (see [82], page 188). After the construction, we will give a proof to show that the construction yields Theorem 3.4.10.

Construction 3.4.11. *Take* (V, B)*, a* $\mathrm{RPBD}(v, M, 1, n+1)$ *with resolution classes* R_∞ *and* $R_1, \dots, R_n$*. Suppose there are integers* $d_i \geq 1$*,* $1 \leq i \leq n$*, such that for each* $b \in R_i$ *there exists an idempotent* $(\mu, |b| - d_i, |b|)$*-latin trade, and integer* $d_\infty \geq 0$ *such that for each* $b \in R_\infty$ *there exists a* $(\mu, |b| - d_\infty, |b|)$*-latin trade. We impose an arbitrary total ordering* $<$ *on* V*. For a block* $b \subseteq V$*, define* b^h *to be the* h*th smallest symbol in* b *when* b *is considered under the ordering imposed on* V*, for* $1 \leq h \leq |b|$*. That is* $\{b^1, \dots, b^{|b|}\} = b$ *and* $b^i < b^{i+1}$ *for* $1 \leq i \leq |b| - 1$*.*

We construct μ $v \times v$ arrays, $T_1, \ldots, T_\mu$, each with rows and columns indexed by V. For each $1 \leq i \leq n$ and each block $b \in R_i$, let $\mathcal{S} = (S_1, \ldots, S_\mu)$ be an idempotent $(\mu, |b|-d_i, |b|)$-latin trade on the set of symbols $\Omega = [|b|]$. For each block $b \in R_\infty$, let $\mathcal{S} = (S_1, \ldots, S_\mu)$ be a $(\mu, |b| - d_\infty, |b|)$-latin trade on the set of symbols $\Omega = [|b|]$. Whenever $(r, c, e) \in S_\alpha$ put (b^r, b^c, b^e) into T_α, for $1 \leq r, c \leq |b|$. Note that if (r, c) is empty in S_α for any $1 \leq r, c \leq |b|$, then (b^r, b^c) is left empty in T_α.

Proof. Consider the μ $v \times v$ arrays $\mathcal{T} = (T_1, \ldots, T_\mu)$ from Construction 3.4.11. We will show that $\mathcal{T}$ is a $(\mu, v + n - \sum_{i=1}^{n} d_i - d_\infty, v)$-latin trade.

During the construction, any single cell $(r, c, e) \in T_\alpha$ with $|\{r, c, e\}| \leq 2$ must have been constructed using some $x \in R_\infty$, as the other blocks were replaced during the construction by idempotent (μ, k, m)-latin trades, which would imply $|\{r, c, e\}| = 3$ by the definition of idempotent μ-way latin trades.

Suppose the construction filled two cells (r_1, c_1, e_1) and (r_2, c_2, e_2) of T_α such that the two cells have two of three indices the same. Let a and b be the values of the two identical indices (for example if the two cells we are observing are $(r, c, e_1), (r, c, e_2) \in T_\alpha$ with $e_1 \neq e_2$, then $a = r$ and $b = c$). If distinct blocks x and y were used respectively to construct (r_1, c_1, e_1) and (r_2, c_2, e_2), then $\{a, b\} \subseteq x \cap y$. By the definition of a PBD$(v, M, 1)$, $|x \cap y| \leq 1$, and so $a = b$. Then $|\{r_1, c_1, e_1\}| \leq 2$ and $|\{r_2, c_2, e_2\}| \leq 2$, and so $x, y \in R_\infty$, which implies $|x \cap y| = 0$ as x and y are distinct blocks in the same resolution class. This forms a contradiction, as $a \in x \cap y$. So any two filled cells that have two of three indices the same were both filled during construction using the same block.

As we filled μ-way latin trades into T_α from these blocks, it follows that no cell was filled twice, each row contains each symbol at most once, and each column contains each symbol at most once. Then each T_α is a partial latin square.

To see $\mathcal{T}$ forms a μ-way latin trade, it is enough to note that $\mathcal{S}(T_\alpha)$ must be the same for each $\alpha \in [\mu]$; that each filled cell (r, c) was filled differently in each T_α, $\alpha \in [\mu]$; and that each row (resp. column) contains setwise the same symbols, each of which are clear from the construction. Then the μ arrays T_α form a μ-way latin trade of order v.

We are left to show $\mathcal{T}$ is $(v + n - \sum_{i=1}^{n} d_i - d_\infty)$-homogeneous. To show that there are $v + n - \sum_{i=1}^{n} d_i - d_\infty$ filled cells in each row, observe that for each symbol r and each resolution class R_i, there is precisely one block b_i such that $r \in b_i \in R_i$, for each i with $1 \leq i \leq n$ or $i = \infty$. Then any filled cells in row r are in the cells $b_i \times b_i$ for some i with $1 \leq i \leq n$ or $i = \infty$. There are $|b_i| - d_i$ filled cells in the intersection of row r and the block of cells $b_i \times b_i$, for $1 \leq i \leq n$ and $i = \infty$, showing row r has a total of $\sum_{i=1}^{n}(|b_i| - d_i) + (|b_\infty| - d_\infty) = v + n - \sum_{i=1}^{n} d_i - d_\infty$ filled cells. The proof is analogous for the number of filled cells per column and for the number of occurrences of each symbol. This shows $\mathcal{T}$ is a μ-way latin trade of order v that is $(v + n - \sum_{i=1}^{n} d_i - d_\infty)$-homogeneous, so we are done. $\square$

Theorem 3.4.12. *Suppose there exists a* $\mathrm{RPBD}(v, M, 1, n+1)$, (V, B) *with resolution classes* $R_1, \ldots, R_n$ *and* R_∞. *Suppose there exists integers* $d_i \geq 1$, $1 \leq i \leq n$, *such that for each* $b \in R_i$ *there exists an idempotent* $(\mu, |b| - d_i, |b|)$*-latin trade and integer* $d_\infty \geq 1$ *such that for each* $b \in R_\infty$ *there exists an idempotent* $(\mu, |b| - d_\infty, |b|)$*-latin trade. Then there exists an idempotent* $(\mu, v + n - \sum_{i=1}^{n} d_i - d_\infty, v)$*-latin trade.*

Proof. Consider the μ $v \times v$ arrays $\mathcal{T} = (T_1, \ldots, T_\mu)$ from Construction 3.4.11 using idempotent $(\mu, |b| - d_i, |b|)$-latin trades to fill in the squares $b \times b$, for blocks $b \in R_\infty$.

The proof of Theorem 3.4.10 shows $\mathcal{T}$ is a $(\mu, v + n - \sum_{i=1}^{n} d_i - d_\infty, v)$-latin trade. We show that $\mathcal{T}$ is idempotent. Assume for the sake of contradiction that $\mathcal{T}$ is not idempotent. Then there exists a $(r, c, e) \in T_\alpha$ with at least two of the three indices r, c, e the same. This only occurs when the block x with $\{r, c, e\} \subseteq x$ and $x \in R_i$ was used along with a non-idempotent $(\mu, k, |x|)$-latin trade to construct the cell $(r, c, e) \in T_\alpha$. But there is no such x as each of the $(\mu, k, |x|)$-latin trades are idempotent. Then $\mathcal{T}$ is idempotent. $\square$

We wish to choose an RPBD$(v, M, 1, n+1)$ with resolution classes $R_1, \ldots, R_n$ and R_∞ such that there will exist idempotent $(\mu, |b| - d_i, |b|)$-latin trades for each $b \in R_i$, for some $d_i \geq 1$ and $i \in \{1, \ldots, n, \infty\}$. By making M contain as few values as possible, we can limit the number of idempotent (μ, k, m)-latin trades that are required to exist, as $|b| \in M$. A resolvable transversal design is a RPBD$(\alpha n, \{\alpha, n\}, 1, n+1)$, and so suits our purposes as $|M| \leq 2$. We are able to modify the resolvable transversal design by removing elements in order to yield RPBD$(v, M, 1, n+1)$ such that v can be any positive integer, while M contains as few values as possible.

Definition 3.4.13. *A* transversal design TD(α, n) *of order n and block size α, is a triple (V, G, B) such that:*

1. *V is a set of αn elements;*
2. *G is a partition of V into α subsets (called the groups), each of size n;*
3. *B is a collection of subsets of V (called the blocks), each of size α ; and*
4. *every unordered pair of elements of V appears in precisely one block of B, or one group of G, but not both.*

Definition 3.4.14. *A* resolvable transversal design RTD(α, n) *of order n and block size α, is a triple (V, G, B) such that B can be partitioned into n resolution classes $R_1, \ldots, R_n$, such that each R_i is a partition of V into n classes.*

The following two lemmata are well known (See III.3.2 and III.3.3 in [33]).

Lemma 3.4.15. *A* RTD(α, n) *is equivalent to a* TD$(\alpha+1, n)$*.*

Lemma 3.4.16. *For n a prime power and $\alpha \leq n$, there exists a* TD$(\alpha+1, n)$ *and hence there exists a* RTD(α, n)*.*

Construction 3.4.17. *Consider a* $\mathrm{RTD}(\alpha, n)$ (V, G, B) *with resolution classes* $R_1, \ldots, R_n$, *and let* $G = \{G_1, \ldots, G_\alpha\}$. *We take* $0 \leq x \leq n$, $0 \leq \gamma \leq \alpha$ *and* $0 \leq u \leq n - x$. *We will form a* $\mathrm{RPBD}(v, M, 1, n+1)$, $(\hat{V}, \hat{B})$, *by deleting a set of* $(n-x)\gamma + u$ *points, which we label as* $\bar{V}$. *The points* $\bar{V}$ *that we delete will be* $n - x$ *points from* G_i *for each* i *with* $\alpha - \gamma + 1 \leq i \leq \alpha$, *and* u *points of* $G_{\alpha-\gamma}$. *Each point that was removed from a group is also removed from any block that contains it. This gives point set* $\hat{V} = V \setminus \bar{V}$, *block set* $\hat{B} = \{b \setminus \bar{V} \mid b \in G \cup B\}$, *and* $n+1$ *resolution classes* $\hat{R}_i = \{b \setminus \bar{V} \mid b \in R_i\}$ *for* $1 \leq i \leq n$ *and* $\hat{R}_\infty = \{b \setminus \bar{V} \mid b \in G\}$.

This results in a $\mathrm{RPBD}(n\alpha - n\gamma + x\gamma - u, M, 1, n+1)$ *with* $M = \{\alpha - (\gamma+1), \ldots, \alpha\} \cup \{x, n-u, n\}$.

It will be useful to summarize the results of this section, which yield the following lemma:

Lemma 3.4.18. *Take* n *a prime power and positive integers* α, x, γ, *and* u *such that* $\alpha \leq n$, $0 \leq x \leq n$, $0 \leq \gamma \leq \alpha$, *and* $0 \leq u \leq n - x$. *Suppose there exists integers* $d_i \geq 1$, $1 \leq i \leq n$ *and* $d_\infty \geq 1$, *such that for each* b *with* $\alpha - (\gamma+1) \leq b \leq \alpha$ *there exists an idempotent* $(3, b - d_i, b)$*-latin trade when* $1 \leq i \leq n$, *and for each* $b \in \{x, n-u, n\}$ *there exists an idempotent* $(3, b - d_\infty, b)$*-latin trade. Then there exists an idempotent* $(3, v + n - \sum_{i=1}^{n} d_i - d_\infty, v)$*-latin trade, where* $v = n\alpha - n\gamma + x\gamma - u$.

Proof. For these values of n, α, x, γ, and u, Lemma 3.4.16 gives us a $\mathrm{RTD}(\alpha, n)$, which we can use in Construction 3.4.17 to yield a resolvable pairwise balanced design that can be used in Theorem 3.4.12 along with the given idempotent $(3, b - d_i, b)$-latin trades that have been assumed to exist, for $i \in \{1, \ldots, n, \infty\}$, to yield the result. □

3.5 Result when $\mu = 3$

We will develop an inductive proof for the existence of idempotent $(3, k, m)$-latin trades for $m > 194$, however we will require the knowledge of the existence of a great deal of base cases. To this end, we will use a computer program to combine the results so far stated in this chapter to deduce the spectrum $\mathcal{IS}_m^3$ for $m \leq 2^{18}$. We will create two computer programs, Program A and Program B, which are implemented in C++ [62].

We begin by finding the spectrum $\mathcal{IS}_m^3$ for $m \leq 5618$ using Program A. The value $5618 = 2 \cdot 53^2$ was chosen as there was some difficulty filling in the spectrum $\mathcal{IS}_{5618}^3$ later on, stemming from the fact that not enough is known about $\mathcal{IS}_{2 \cdot 53}^3$, and so Lemma 3.4.2 cannot be used to fill in the spectrum $\mathcal{IS}_{5618}^3$. We split the computation into four parts. Parts 1, 2, and 4 are straightforward to program, but there is some complications with Part 3. We begin with a $(5618+1) \times (5618+1)$ array of booleans $A = [a_{k,m}]$, where we set $a_{k,m} = false$ for each $0 \leq k, m \leq 5618$. When we find that there does exist an idempotent $(3, k, m)$-latin trade, we set $a_{k,m} = true$.

PART 1: As there trivially exists an idempotent $(3, 0, m)$-latin trade, we set $a_{0,m} = true$ for each $0 \leq m \leq 5618$. As the existence of idempotent $(3, k, m)$-latin trades in Theorem 3.3.5, Theorem 3.3.10, Theorem 3.4.1, and Theorem 3.4.7 do not depend on the existence of smaller idempotent $(3, k', m')$-latin trades, we set $a_{k,m} = true$ for these values.

We can use the idempotent $(2, k_1, 2m' + 1)$-latin trades of Theorem 3.3.9 when $3 \leq k_1 < 2m' + 1$, $2m' + 1 \geq 5$, and $(k, 2m' + 1) \neq (3, 5)$, along with the $(2, k_2, m_2)$-latin trades of Theorem 3.2.5 for $2 \leq k_2 \leq m_2$, with $k_i = 2$ only if m_i is even, with Theorem 3.3.2 to yield an idempotent $(3, k_1k_2, (2m' + 1)m_2)$-latin trade. This does not depend on the existence of smaller idempotent $(3, k, m)$-latin trades, so we set $a_{k_1k_2,(2m'+1)m_2} = true$ under these conditions.

PART 2: Theorem 3.3.1 and Theorem 3.3.3 each require the knowledge of the existence of smaller idempotent $(3, k, m)$-latin trades, however we can gather this information from what we have stored in A. Theorem 3.3.3 also uses the existence of non-idempotent $(3, m, m)$-latin trades, and a non-idempotent $(3, 5, 6)$-latin trade (an example of a $(3, 5, 6)$-latin trade is shown in the next section).

PART 3: Programming Lemma 3.4.18 is not completely straightforward, as the time required can be quite large if not done with due care. We implement Lemma 3.4.18 twice. The first implementation uses $d_\infty = 1$, and the second implementation uses $\gamma = 0$. Both of these restrictions speed up the computation immensely, and the values not covered by one are covered by the other. By first looping over α and γ, we can store the values of d_i such that there exists an idempotent $(3, b - d_i, b)$-latin trade for each b with $\alpha - \gamma - 1 \leq b \leq \alpha$. Then we are able to find the possible values of $\sum_{i=1}^{n} d_i$ without much extra computation as we increase n, by storing the previously computed values of $\sum_{i=1}^{n-1} d_i$.

PART 4: We once again apply the procedure for Theorem 3.3.1, which fills in a couple of gaps in the spectrum introduced incidentally in Part 3.

Performing this computation gives the following lemma:

Lemma 3.5.1. *For $14 \leq m \leq 5618$, there exists idempotent $(3, k, m)$-latin trades for $5 \leq k \leq m$ except, perhaps, for those values in Table 3.1.*

We need to extend the base results further, which we achieve by way of another computer program, Program B. We begin with an array of $2^{18} + 1$ booleans $B = [b_m]$, where we set $b_m = false$ for each $0 \leq m \leq 2^{18}$. When we find that there exists idempotent $(3, k, m)$-latin trades for every $5 \leq k \leq m$, we set $b_m = true$. We begin by setting the values of b_m to be true when the values $a_{k,m}$ each are true for $5 \leq k \leq m$. Then Lemma 3.4.2 with $y = 5$ tells us that we can set b_{nm} to be true whenever $m \geq 10$, b_m is true, $n \geq 3$, and if $n = 6$ then $m \geq 20$.

In the case that m is a prime or twice a prime, we can apply Theorem 3.4.1 and Theorem

3.4.7 to yield idempotent $(3, k, m)$-latin trades for $5 \leq k \leq l_1$, and we can apply Lemma 3.4.18 with $x = 7$ to yield idempotent $(3, k, m)$-latin trades for $l_2 \leq k \leq m$, for some integers l_1, l_2. To save computation, we only consider Lemma 3.4.18 with $\gamma \in \{0, 1\}$ and $d_\infty = 1$. Note that this means $(3, b - d_\infty, b)$-latin trades always exist for $b \geq 7$, as stated in Theorem 3.3.5. Our program checks if $b_{\alpha-2} = b_{\alpha-1} = b_\alpha = true$ for each $\alpha \geq 16$ and $n \geq \alpha$. In this case, we assume $d_i \leq \alpha - 7$. Then the conditions of a $(3, \alpha - (\gamma + 1) - d_i, \alpha - (\gamma + 1))$-latin trade existing hold independently of whether $\gamma = 0$ or $\gamma = 1$. Then it will be more convenient to write $u' = \gamma(n - 7) + u$. If so, for each $m = n\alpha - u'$ with $n \geq \alpha$, $0 \leq u' \leq 2(n - 7)$, and m a prime or twice a prime, we know that there exists an idempotent $(3, k, m)$-latin trade for each k with $m + n - n(\alpha - 7) - 1 = 8n - u' - 1 \leq k \leq m - 1$. To find the existence of idempotent $(3, k, m)$-latin trades with $5 \leq k < 8n - u' - 1$, we find the greatest λ with $m = \lambda(\lambda + a) + b$, $a > b$, $\gcd(m, \lambda) = 1$, and $\lambda \geq 5$. Then Theorem 3.4.7 yields the existence of idempotent $(3, k, m)$-latin trades with $18 \leq k \leq \lambda^2$. There exists idempotent $(3, k, m)$-latin trades for $5 \leq k \leq 17$ by Theorem 3.4.1. If $8n - u' - 1 \leq \lambda^2$, then there exists an idempotent $(3, k, m)$-latin trade for $5 \leq k \leq m - 1$, and so we set b_m to be true.

Performing this computation gives the following lemma:

Lemma 3.5.2. *For $14 \leq m \leq 2^{18}$, there exists idempotent $(3, k, m)$-latin trades for $5 \leq k \leq m$ except, perhaps, for those values in Table 3.1.*

We have been able to apply Lemma 3.4.18 in this computation as we have been able to run a procedure to check which integers n are prime powers. In order to create a theoretic construction, we restrict the prime powers that we use, so that n is of the form 2^p, for an integer p. We are then able to show, despite this restriction, that Lemma 3.4.18 can yield a large portion of the spectrum of $(3, k, m)$-latin trades for all $m \geq 2^{18}$.

Lemma 3.5.3. *Take $p \geq 10$. Suppose there exists idempotent $(3, k', m')$-latin trades for $5 \leq k' \leq m' - 1$ and $2^{p-2} - 6 \leq m' \leq 2^p$. Then there exists idempotent $(3, k, m)$-latin* *trades for* $14 \cdot 2^p \leq k \leq m$ *and* $2^{2p-2} < m \leq 2^{2p}$*.*

Proof. Take $\alpha = n = 2^p$, $p \geq 10$, $0 \leq \gamma \leq \gamma_{\max}$, $\gamma_{\max} = 2^p - 2^{p-2} + 5$, $x = 7$, and $0 \leq u \leq n - 7$. We assume the existence of idempotent $(3, k', m')$-latin trades when $5 \leq k' \leq m' - 1$ and $\alpha - (\gamma_{\max} + 1) \leq m' \leq \alpha$, noting that $\alpha - (\gamma_{\max} + 1) = 2^{p-2} - 6$. There exists a $(3, m' - 1, m')$-latin trade for $m' \in \{7, n - u, n\}$ by Theorem 3.3.5, as $m' \geq 7$.

Then Lemma 3.4.18 with these idempotent $(3, k', m')$-latin trades yields an idempotent $(3, m + n - \sum_{i=1}^{n} d_i - d_\infty, m)$-latin trade with $m = n\alpha - \gamma(n - 7) - u$, where $1 \leq d_i \leq \alpha - (\gamma + 1) - 5$ for $1 \leq i \leq n$, and $d_\infty = 1$. Taking $d_i \leq \alpha - (\gamma + 1) - 5$ assures us that $m' - d_i \geq 5$ for each m' with $\alpha - (\gamma + 1) \leq m' \leq \alpha$, and so an idempotent $(3, m' - d_i, m')$-latin trade exists by our assumptions.

Then this procedure yields an idempotent $(3, k, m)$-latin trade for each $k \in \{m+n-n(\alpha-(\gamma+1)-5)-1, \ldots, m-1\} = \{7\gamma+7n-u-1, \ldots, m-1\} \supseteq \{14n, \ldots, m-1\}$, which holds for each $m = n\alpha-\gamma(n-7)-u$ within $n\alpha-(\gamma_{\max}+1)(n-7) \leq m \leq n\alpha$. In particular, it holds that $n\alpha-(\gamma_{\max}+1)(n-7) = 2^{2p}-(2^p-2^{p-2}+6)(2^p-7) = 2^{p-2}-3\cdot 2^{p-2}+42 \leq 2^{2p-2}$ and so there exists an idempotent $(3, k, m)$-latin trade for $2^{2p-2} \leq m \leq 2^{2p}$ and $14\cdot 2^p \leq k \leq m-1$, showing the result. ☐

Theorem 3.5.4. *For* $m \geq 5$*, there exists an idempotent* $(3, k, m)$*-latin trade for* $5 \leq k \leq m-1$ *except possibly for those values in Table 3.1 and when* $(k, m) \in \{(5, 6), (5, 10), (5, 13)\}$*.*

Proof. Define $P(r)$ to be the statement "There exists idempotent $(3, k, m)$-latin trades for each $5 \leq k \leq m - 1$ and $2^{r-1} - 6 \leq m \leq 2^r$".

Lemma 3.5.2 shows $P(r)$ is true for $9 \leq r \leq 18$. Assume for the sake of strong induction that $P(r)$ is true for $9 \leq r < R$, with $R \geq 19$. Then $P(\lceil \frac{R}{2} \rceil - 1)$ and $P(\lceil \frac{R}{2} \rceil)$ are true, as $9 \leq \lceil \frac{R}{2} \rceil - 1 < R$. This makes the premise of Lemma 3.5.3 with $p = \lceil \frac{R}{2} \rceil$ true, and so there exists an idempotent $(3, k, m)$-latin trade for $14 \cdot 2^{\lceil \frac{R}{2} \rceil} \leq k \leq m - 1$ and $2^{R-1} \leq m \leq 2^R$, as $2^{2\lceil\frac{R}{2}\rceil-2} \leq 2^{R-1}$ and $2^R \leq 2^{2\lceil\frac{R}{2}\rceil}$. As $P(R-1)$ is true, we can apply Theorem 3.3.4 with $k' = 2^{R-2}$ and for each $k \in \{5, \dots, 2^{R-2}-1\}$, which yields idempotent $(3,k,m)$-latin trade for $5 \leq k \leq 2^{R-2}-1$ and $2^{R-1} \leq m \leq 2^R$. As $14 \cdot 2^{\lceil\frac{R}{2}\rceil} \leq 2^{R-2}-1$, this shows there exists an idempotent $(3,k,m)$-latin trade for $5 \leq k \leq m-1$ and $2^{R-1} \leq m \leq 2^R$, and so $P(R)$ is true. By strong induction, $P(r)$ is true for $r \geq 9$. Theorem 3.4.1 and Lemma 3.5.2 complete the result when $5 \leq m < 2^8 - 6$. □

3.6 Results

As $\mathcal{IS}_m^\mu \subseteq \mathcal{S}_m^\mu$, the results of Theorem 3.5.4 also yield the existence of $(3,k,m)$-latin trades for identical values of k and m. There are a few more non-idempotent $(3,k,m)$-latin trades that we can find.

There does not exist a large set of idempotent latin squares of order $n = 6$, however there does exist a $(4,5,6)$-latin trade given by:

$(2,3,4,5)$	$\bullet$	$(1,4,5,3)$	$(5,2,1,4)$	$(3,5,2,1)$	$(4,1,3,2)$
$\bullet$	$(3,2,5,4)$	$(6,3,4,5)$	$(4,5,2,6)$	$(5,6,3,2)$	$(2,4,6,3)$
$(1,4,5,3)$	$(4,5,3,6)$	$(5,1,6,4)$	$\bullet$	$(6,3,1,5)$	$(3,6,4,1)$
$(4,5,2,1)$	$(5,6,4,2)$	$\bullet$	$(1,4,6,5)$	$(2,1,5,6)$	$(6,2,1,4)$
$(5,1,3,2)$	$(6,3,2,5)$	$(3,5,1,6)$	$(2,6,5,1)$	$(1,2,6,3)$	$\bullet$
$(3,2,1,4)$	$(2,4,6,3)$	$(4,6,3,1)$	$(6,1,4,2)$	$\bullet$	$(1,3,2,6)$

Here, the partial latin squares have been concatenated, so that cell (r,c) has been filled with the ordered 4-tuple given by $(t_1(r,c), t_2(r,c), t_3(r,c), t_4(r,c))$, where the four partial latin squares $T_i = [t_i(r,c)]$, $1 \le i \le 4$, form the $(4,5,6)$-latin trade. Then this yields a non-idempotent $(3,5,6)$-latin trade.

Applying Theorem 7 of [7] to the combination of a $(3,5,6)$-latin trade and a $(3,5,7)$-latin trade yields a $(3,5,13)$-latin trade. There exists a $(3,m,m)$-latin trade for $m \ge 3$ by Theorem 3.2.6, and a $(3,m,2m)$-latin trade by applying Theorem 7 of [7] to the combination of two $(3,m,m)$-latin trades. Then the primary result of this chapter, combined with previous results [7], can be written as the following theorem:

Theorem 3.6.1. *For $m \ge 4$ there exists a $(3,k,m)$-latin trade for $4 \le k \le m$ except, perhaps, for those unstared values in Table 3.1 and for $(k,m) = (4,11)$, and except for those values with $(k,m) \in \{(4,6),(4,7)\}$. For $m \ge 3$, there exists a $(3,3,m)$-latin trade only when $3|m$.*

This leaves us with 194 exceptions for which we do not know if a $(3,k,m)$-latin trade exists, and 2 exceptions when we know that there does not exist a $(3,k,m)$-latin trade.

3.7 Future work

Given the relative success of finding base rows from Theorem 3.4.1, where the program terminated rather early within the search space, it seems reasonable that $(3,k,m)$-latin trades with values in Table 3.1 could exist, and we can use this as evidence towards a conjecture:

Conjecture 3.7.1. *There exists a $(3,k,m)$-latin trade exactly when $k = 3$ and $3|m$, and when $4 \le k \le m$, except in the cases that $(k,m) \in \{(4,6),(4,7),(4,11)\}$.*

It also seems that similar techniques used in this chapter could be used to fill in the spectrum of $(4,k,m)$-latin trades. In addition, it may be of interest to investigate the spectrum of circulant (μ,k,m)-latin trades.

m	k
22	19
23	$19, 20, 21$
26	$19, 21, 23$
29	$19, 20, 21, 22, 23, 24, 25, 26, 27$
31	$19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29$
34	$19, 21, 23, 25, 27, 29, 31$
37	$33, 34, 35$
38	$19^*, 21, 23, 25, 27, 29, 31, 33, 35$
41	$35, 36, 37, 38, 39$
43	$36, 37, 38, 39, 40, 41$
46	$27, 29, 31, 33, 35, 37, 39, 41, 43$
53	$50, 51$
58	$29^*, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55$
59	$55, 56, 57$
62	$31^*, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59$
74	$35, 37^*, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71$
82	$41^*, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65$
86	$51, 53, 55, 57, 59, 61, 63, 65, 67, 69$
94	$51, 53, 55, 57, 59, 61, 63, 65$
106	$53^*, 55, 57, 59, 61, 63, 65, 67, 69$
122	$59, 61, 63, 65, 67, 69, 71, 73, 75, 77$
134	$67^*, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93$
146	$83, 85, 87, 89, 91, 93, 95, 97, 99, 101$
158	$83, 85, 87, 89, 91$
194	$97^*, 99, 101$

Table 3.1: Values where no idempotent $(3, k, m)$-latin trade is known to exist with $6 \leq k \leq m - 1$. The starred values indicate when a non-idempotent $(3, k, m)$-latin trade is known to exist.

A μ-way latin trade $(Q_1, \ldots, Q_\mu)$ can be said to be *primary* if there is no μ-way latin trade $(R_1, \ldots, R_\mu)$ such that $R_\alpha \subsetneq Q_\alpha$. A μ-way latin trade is said to be *minimal* if there is no partial latin square $R \subsetneq Q_1$ such that there exists a 2-way trade (R, R'). Primary $(2, k, m)$-latin trades were conjectured to exist for $3 \leq k \leq m$ in [30]. It would be of interest to investigate primary and minimal (μ, k, m)-latin trades in the future.

Chapter 4

Enumeration of MNOLS

4.1 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 [58]). 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 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.

Based upon the significance and usefulness that is exhibited in the study of MOLS, Raghavarao, Shrikhande, and Shrikhande [76] introduced a modification to the definition of orthogonality to overcome restrictions for the even order case. Recall that 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 μ latin squares of order n that are pairwise nearly orthogonal, which are denoted as collections of μ 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 [45] and [77]. A similar technique, called canonical augmentation [67], has been used to generate latin rectangles by augmenting a row at a time (see also [57][71]). This is not the only method of enumerating latin rectangles, and a variety of enumerative techniques have been applied to solve it (see [72] and the citations contained within). Recently, this work has lead to the enumeration of MOLS for order less than or equal to 9 [42]. In a similar vein, we will perform three orderly algorithms that generate collections of cyclic μ MNOLS of order n under several equivalences. See [55] for a general reference on this kind of enumeration problem.

The pioneering work [76] on sets of μ MNOLS of order n investigated an upper bound on μ when n is fixed, and they showed that if there exists a set of μ 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 μ MNOLS of order n obtains this bound, it is called a complete set of μ MNOLS of order n.

The authors proceeded to explore the existence of sets of μ MNOLS of order n by investigating μ cyclic MNOLS of order n; that is each latin square L has $(r, c, e) \in L$ if and only if $(r, c+1, e+1) \in L$ for all $r, c \in [0, n-1]$, recalling that the entries are taken mod n. The sets of μ 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 [75] that there does not exist four MNOLS of order 6, and so the bound is not tight for $n = 6$.

Further results [61] 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 μ cyclic MNOLS of order n called isotopic equivalence (details in Section 4.2). They found the number of isotopically non-equivalent sets of μ cyclic MNOLS of order n for $n \leq 12$. The number of these sets of μ cyclic MNOLS of order n is given in Table 1.1.

The existence of three cyclic MNOLS of orders $48k+14$, $48k+22$, $48k+38$, and $48k+46$ when $k \geq 0$ was documented in [35], and also in [36], 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 chapter, we find the maximum μ such that there exists a set of μ cyclic MNOLS of order n for $n \leq 16$, as well as providing a full enumeration of sets and lists of μ cyclic MNOLS of order n under a variety of equivalences with $n \leq 16$. This will resolve in the negative a conjecture of [61] that proposed the maximum μ for which a set of μ cyclic MNOLS of order n exists is $\lceil n/4 \rceil + 1$ (the maximum μ appears erroneously as $\lceil n/8 \rceil + 1$ in the original conjecture [78], and the maximum value we have written was the intended conjecture).

4.2 Further definitions

We will enumerate both ordered lists and unordered sets of μ MNOLS of order n. A *set of μ MNOLS* of order n is a set $\{L_1, \dots, L_\mu\}$ such that L_i, L_j are nearly orthogonal for $1 \leq i, j \leq \mu$, $i \neq j$. A *list of μ MNOLS* of order n is an ordered list $(L_1, \dots, L_\mu)$ 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 μ MNOLS of order n. We will write *collection* when a statement holds for either a list or set. A list of μ MNOLS of order n $(L_1, \dots, L_\mu)$ is *reduced* if L_1 has its first row and column in natural order.

We assume the 3 components of such a triple are taken mod n, so that $(r, c, e + n) = (r, c + n, e) = (r + n, c, e)$. We also write $L(r, c) = e$ when $(r, c, e) \in L$.

Define an ordering $\triangleleft$ on the set of all latin squares of order n as follows: for two latin squares, L and M, we have $L \triangleleft M$ 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 $\triangleleft$ on lists on μ MNOLS of order n to be such that for two lists of μ MNOLS of order n, $\mathcal{L} = (L_1, \ldots, L_n)$ and $\mathcal{M} = (M_1, \ldots, M_n)$, we have $\mathcal{L} \triangleleft \mathcal{M}$ if either $\mathcal{L} = \mathcal{M}$ or there is $1 \leq \beta \leq \mu$ such that $L_\beta \triangleleft M_\beta$ and $L_{\beta'} = M_{\beta'}$ for $\beta' < \beta$. Such an ordering is known as dictionary ordering when we consider a list of μ latin squares of order n to be an element of $([n]^3)^{n^2 \cdot \mu}$.

Given a group G and set Ω, a function $h : \Omega \times G \to \Omega$ is a *group action* if for all $\alpha \in \Omega$ and $a, b \in G$, $h(h(\alpha, a), b) = h(\alpha, ab)$ and $h(\alpha, e) = \alpha$, where $e \in G$ is the identity element of G. Then Ω is a *G-space* when equipped with such a group action. The *orbit* of $\alpha \in \Omega$ is $\alpha \cdot G = \{h(\alpha, g) \mid g \in G\}$, and the stabilizer of $\alpha \in \Omega$ is $G_\alpha = \{g \in G \mid h(\alpha, g) = \alpha\}$. If there is just one orbit, Ω is a *transitive* G-space.

Consider the group actions on group $G = \mathcal{S}_n^3$ (the direct product $\mathcal{S}_n \times \mathcal{S}_n \times \mathcal{S}_n$) and set Ω, where Ω is a set of collections of latin squares of order n. Given a $(\sigma_R, \sigma_C, \sigma_E) \in \mathcal{S}_n^3$ and a collection of latin squares $\mathcal{L} \in \Omega$, we define $h : \Omega \times \mathcal{S}_n^3 \to \Omega$ by setting $h(\mathcal{L}, (\sigma_R, \sigma_C, \sigma_E))$ to be the collection of latin squares that is obtained by uniformly permuting the rows (resp. columns, symbols) of the latin squares in $\mathcal{L}$ by σ_R (resp. σ_C, σ_E). Clearly Ω must be defined so that $h : \Omega \times G \to \Omega$. As a form of shorthand, we write $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ instead of $h(\mathcal{L}, (\sigma_R, \sigma_C, \sigma_E))$.

Two lists of μ MNOLS of order n, $\mathcal{L}$ and $\mathcal{N}$, are *list-isotopic* if there exists $(\sigma_R, \sigma_C, \sigma_E) \in$

$\mathcal{S}_n^3$ with $\mathcal{N} = \mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$. Two sets of μ MNOLS of order n are *isotopic* if some ordering of the latin squares within each set gives two list-isotopic lists of μ MNOLS of order n. Two lists of μ MNOLS of order n, $\mathcal{L}$ and $\mathcal{N}$, are *set-isotopic* if forgetting the order on the lists gives two isotopic sets of μ MNOLS of order n. Clearly the number of sets of μ MNOLS of order n that are distinct up to isotopy is the same as the number of lists of μ MNOLS of order n up to set-isotopy.

4.3 Cyclic MNOLS

We take $I = I_n \in \mathcal{S}_n$ to be the identity permutation of order n, $\tau = \tau_n \in \mathcal{S}_n$ 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 $m_x = m_{x,n} \in \mathcal{S}_n$ 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 describe those $(\sigma_R, \sigma_C, \sigma_E) \in \mathcal{S}_n^3$ such that given a collection of cyclic MNOLS $\mathcal{L}$, $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ is also a collection of cyclic MNOLS. For the following, recall that each component of a triple $(r, c, e) \in L$ is taken modulo n.

Taking a collection of μ MNOLS of order n, $\mathcal{L}$, if $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ is a collection of μ MNOLS of order n then it is immediate clear that $\sigma_E(y+n/2) = \sigma_E(y)+n/2$ for all $0 \leq y \leq n/2-1$, in order to preserve near orthogonality. The converse is also true:

Lemma 4.3.1. *Consider a collection of* μ MNOLS *of order* n*,* $\mathcal{L}$*, along with* $(\sigma_R, \sigma_C, \sigma_E) \in \mathcal{S}_n^3$ *such that* $\sigma_E(y+n/2) \equiv \sigma_E(y)+n/2 \mod n$*, for all* $0 \leq y < n/2$*. Then* $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ *is also a collection of* μ 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 σ_R, σ_C, and σ_E. Consider the multisets $P = \{(e, e') \mid (r, c, e) \in L_1 \text{ and } (r, c, e') \in L_2 \text{ for } r, c \in \{0, \ldots, n-1\}\}$ and also $P' =$

$\{(e,e') \mid (r,c,e) \in L'_1$ and $(r,c,e') \in L'_2$ for $r,c \in \{0,\ldots,n-1\}\}$. 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 μ MNOLS of order n.

□

Permuting the rows in a collection of cyclic MNOLS does not destroy the cyclic property:

Lemma 4.3.2. *Consider a collection of μ* MNOLS *of order n, $\mathcal{L}$, and permutation $\sigma_R \in \mathcal{S}_n$. If $\mathcal{L}$ is cyclic, then so is $\mathcal{L}(\sigma_R, I, I)$.*

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 σ. 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 $\mathcal{L}(\sigma_R, I, I)$ is cyclic. □

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

Lemma 4.3.3. *Consider a collection of μ cyclic* MNOLS *of order n, $\mathcal{L}$. Then $\mathcal{L} = \mathcal{L}(I, \tau^i, \tau^i)$, for $0 \le i < n$, 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 τ^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. $\square$

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

Lemma 4.3.4. *Consider a collection of μ MNOLS of order n, $\mathcal{L}$, and permutations $\sigma_C, \sigma_E \in \mathcal{S}_n$ such that $\sigma_E(y + n/2) \equiv \sigma_E(y) + n/2 \mod n$, for all $0 \leq y < n/2$. If both $\mathcal{L}$ and $\mathcal{L}(I, \sigma_C, \sigma_E)$ are cyclic, then $\sigma_C = m_x \cdot \tau^{\sigma_C(0)}$ and $\sigma_E = m_x \cdot \tau^{\sigma_E(0)}$, 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 cell $(r', c', e') \in L$ if and only if cell $(r', \sigma_C(c'), \sigma_E(e')) \in L'$. 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 $\sigma_E = m_x \cdot \tau^{\sigma_E(0)}$. Since σ_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 $\sigma_C = m_x \cdot \tau^{\sigma_C(0)}$.

$\square$

These facts come together to tell us exactly the form of any $(\sigma_R, \sigma_C, \sigma_E) \in \mathcal{S}_n^3$ such that both $\mathcal{L}$ and $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ are cyclic MNOLS:

Lemma 4.3.5. *Consider collection of μ cyclic MNOLS of order n, $\mathcal{L}$ and $\mathcal{N}$. Then*

$\mathcal{N} = \mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ if and only if there exists integers $0 \leq x, j' < n$ with $\gcd(x, n) = 1$, such that $\mathcal{N} = \mathcal{L}(\sigma_R, m_x, m_x \cdot \tau^{j'})$.

Proof. The reverse implication is clear, so we show the forward implication. Because $\mathcal{N} = \mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ is cyclic, then $\mathcal{L}(I, \sigma_C, \sigma_E)$ is cyclic by Lemma 4.3.2, which by Lemma 4.3.4 implies $\sigma_C = m_x \cdot \tau^{\sigma_C(0)}$ and $\sigma_E = m_x \cdot \tau^{\sigma_E(0)}$ for some x with $\gcd(x, n) = 1$. But then $\mathcal{N} = \mathcal{L}(\sigma_R, m_x \cdot \tau^{\sigma_C(0)}, m_x \cdot \tau^{\sigma_E(0)}) = \mathcal{L}(\sigma_R, m_x, m_x \cdot \tau^{\sigma_E(0) - \sigma_C(0)})$ by Lemma 4.3.3 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 [61] on isomorphisms of $(t, 2m)$-difference sets, which are equivalent to set-isotopisms of t MNOLS of order $2m$.

4.4 Group actions for cyclic MNOLS

Let C_n^μ be the set of all lists of μ cyclic MNOLS of order n. From the previous section, not every $(\sigma_R, \sigma_C, \sigma_E) \in \mathcal{S}_n^3$ and $\mathcal{L} \in C_n^\mu$ has $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E) \in C_n^\mu$. In fact, Lemma 4.3.5 informs us of those $(\sigma_R, \sigma_C, \sigma_E)$ that can be used.

Given a list of μ cyclic MNOLS of order n, $\mathcal{L} = (L_1, \ldots, L_\mu)$, define the permutations $\overline{m}_x, \overline{\tau}, r_\sigma : C_n^\mu \to C_n^\mu$ by $\mathcal{L} \cdot \overline{m}_x = \mathcal{L}(I, m_x, m_x)$, $\mathcal{L} \cdot \overline{\tau} = \mathcal{L}(I, I, \tau)$, and $\mathcal{L} \cdot r_\sigma = \mathcal{L}(\sigma, I, I)$, for $\sigma \in \mathcal{S}_n$, $1 \leq x \leq n-1$, and $\gcd(x, n) = 1$. When $\mathcal{L}$ is cyclic, so is each of $\mathcal{L} \cdot \overline{m}_x$, $\mathcal{L} \cdot \overline{\tau}$, and $\mathcal{L} \cdot r_\sigma$ by Lemma 4.3.5. Further, Lemma 4.3.5 yields that if $\mathcal{L}$ and $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E)$ are both cyclic MNOLS, there always exists $\sigma \in \mathcal{S}_n$ and $1 \leq j, x \leq n$ with $\gcd(x, n) = 1$ such that $\mathcal{L}(\sigma_R, \sigma_C, \sigma_E) = ((\mathcal{L} \cdot r_\sigma) \cdot \overline{\tau}^j) \cdot \overline{m}_x$.

Define $M = \{\overline{m}_x \mid 1 \leq x \leq n-1 \text{ and } \gcd(x, n) = 1\}$, $T = \{\tau^i \mid 0 \leq i \leq n-1\}$, and $R = \{r_\sigma \mid \sigma \in \mathcal{S}_n\}$, each of which form a group under composition.

Given a list of μ cyclic MNOLS of order n, $\mathcal{L} = (L_1, \ldots, L_\mu)$, we define the permutations

$s_\sigma : C_n^\mu \to C_n^\mu$ by $\mathcal{L} \cdot s_\sigma = (L_{\sigma(1)}, \dots, L_{\sigma(\mu)})$, for $\sigma \in \mathcal{S}_\mu$. Let $S = \{s_\sigma \mid \sigma \in \mathcal{S}_\mu\}$, which forms a group under composition. The orbit of these permutations on $\mathcal{L}$, $\mathcal{L} \cdot S$, contains as its elements any list of μ MNOLS of order n that becomes the set $\{L_1, \dots, L_\mu\}$ when we forget the ordering on the list.

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

From here, we will just consider the group actions $h : C_n^\mu \times G \to C_n^\mu$, where G is any subgroup of $\langle M, T, R, S \rangle$. As C_n^μ is a G-space, it can be uniquely expressed as a disjoint union of transitive G-spaces [18], specifically C_n^μ is the disjoint union of the orbits $\{\mathcal{L} \cdot G \mid \mathcal{L} \in C_n^\mu\}$, each of which is a transitive G-space.

The orbits $\mathcal{L} \cdot \langle M, T, R, S \rangle$ for $\mathcal{L} \in C_n^\mu$ are called the *set-isotopy* classes of $\mathcal{C}_n^\mu$, and two lists in the same set-isotopy class are called *set-isotopic*. The stabilizer $\mathrm{Is}_s(\mathcal{L}) = \{\alpha \in \langle M, T, R, S \rangle \mid \mathcal{L} \cdot \alpha = \mathcal{L}\}$ is called the *set-autotopy* group of $\mathcal{L}$, and each contained element is called a *set-autotopy* of $\mathcal{L}$. The number of distinct orbits $\mathcal{L} \cdot \langle M, T, R, S \rangle$ over all $\mathcal{L} \in C_n^\mu$ is the number of sets of μ MNOLS of order n distinct up to isotopy. Note that $\langle M, T, R, S \rangle$ has order $\phi(n) \cdot n \cdot n! \cdot \mu!$.

We will also be interested in the subgroups $\langle M, T, R \rangle \leq \langle M, T, R, S \rangle$, $\langle R, S \rangle \leq \langle M, T, R, S \rangle$, $\langle R \rangle \leq \langle M, T, R, S \rangle$, and $\langle S \rangle \leq \langle M, T, R, S \rangle$. Their orbits are called, respectively, the list-isotopy classes, set-reduced classes, the list-reduced classes, and the set classes of $\mathcal{C}_n^\mu$. We call the stabilizers of $\mathcal{L} \in \mathcal{C}_n^\mu$, respectively, $\mathrm{Is}_l(\mathcal{L})$, $\mathrm{Red}_s(\mathcal{L})$, $\mathrm{Red}_l(\mathcal{L})$, and $\mathrm{Set}(\mathcal{L})$. 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 μ 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 μ cyclic MNOLS of order n;
2. the number of isotopy classes of lists of μ cyclic MNOLS of order n;
3. the number of reduced sets of μ cyclic MNOLS of order n;
4. the number of reduced lists of μ cyclic MNOLS of order n;
5. the number of sets of μ cyclic MNOLS of order n;
6. the number of lists of μ cyclic MNOLS of order n;

For any $\mathcal{L} \in C_n^{\mu}$, both $\mathrm{Red}_l(\mathcal{L})$ and $\mathrm{Set}(\mathcal{L})$ contain only the identity permutation $\mathcal{L} \cdot e = \mathcal{L}$, hence $|\mathrm{Red}_l(\mathcal{L})| = 1$ and $|\mathrm{Set}(\mathcal{L})| = 1$, and so the orbit $\mathcal{L} \cdot \langle R \rangle$ has size $n!$ and the orbit $\mathcal{L} \cdot \langle S \rangle$ has size $\mu!$, independent of the choice of $\mathcal{L}$. Each set-isotopy class can be partitioned 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 μ 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 4.4.1. *Given $\mathcal{L} \in \mathcal{C}_n^{\mu}$:*

1. *the number of list-isotopy classes within the set-isotopy class of $\mathcal{L}$ is:*
   $\mu! \cdot |\mathrm{Is}_l(\mathcal{L})|/|\mathrm{Is}_s(\mathcal{L})|$;
2. *the number of set-reduced classes within the set-isotopy class of $\mathcal{L}$ is:*
   $\phi(n) \cdot n \cdot |\mathrm{Red}_s(\mathcal{L})|/|\mathrm{Is}_s(\mathcal{L})|$;
3. *the number of list-reduced classes within the set-isotopy class of $\mathcal{L}$ is:*
   $\phi(n) \cdot n \cdot \mu! \cdot |\mathrm{Red}_l(\mathcal{L})|/|\mathrm{Is}_s(\mathcal{L})| = \phi(n) \cdot n \cdot \mu!/|\mathrm{Is}_s(\mathcal{L})|$;

4. *the number of set classes within the set-isotopy class of* $\mathcal{L}$ *is:*
   $\phi(n) \cdot n \cdot n! \cdot |\mathrm{Set}(\mathcal{L})|/|\mathrm{Is}_s(\mathcal{L})| = \phi(n) \cdot n \cdot n!/|\mathrm{Is}_s(\mathcal{L})|$; *and*

5. *the number of lists within the set-isotopy class of* $\mathcal{L}$ *is:*
   $\phi(n) \cdot n \cdot \mu! \cdot n!/|\mathrm{Is}_s(\mathcal{L})|$.

Proof. By the orbit-stabilizer theorem. □

4.5 Canonical forms

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

Typically, enumerations involving latin squares [54][72] 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 [70] 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 μ 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 μ 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 μ MNOLS of order n that are not canonical.

Let $\mathcal{M}_i \in C_i$ be such that $\mathcal{M}_i \lhd \mathcal{L}$ for each $\mathcal{L} \in C_i$. In this chapter we will consider the canonical form f defined by $f(\mathcal{L}) = \mathcal{M}_i$, for $\mathcal{L} \in C_i$. For example if we take C_i to be the list-reduced classes, the canonical representation of each class is the unique list of μ MNOLS of order n amongst the class that is reduced.

This aligns with our previous definition of reduced lists of μ 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 set-reduced classes, we call the canonical representation of each class *set-reduced*. Similarly we say a list is *set-canonical* if we take the partition of C_n^μ into set-isotopy classes, and *list-canonical* if we take the partition of C_n^μ into list-isotopy classes.

Lemma 4.5.1. *Given the partition of $\mathcal{C}_n^\mu$ into set-isotopy classes and f a canonical form:*

1. *The number of set-isotopy classes in $\mathcal{C}_n^\mu$ is $|\mathrm{Im}(f)|$.*

2. *The number of list-isotopy classes in $\mathcal{C}_n^\mu$ is $\sum_{\mathcal{L}\in \mathrm{Im}(f)} \mu! \cdot |Is_l(\mathcal{L})|/|Is_s(\mathcal{L})|$.*

3. *The number of set-reduced classes in* $\mathcal{C}_n^\mu$ *is* $\sum_{\mathcal{L}\in \mathrm{Im}(f)} \phi(n) \cdot n \cdot \frac{|\mathrm{Red}_s(\mathcal{L})|}{|Is_s(\mathcal{L})|}$.

4. *The number of list-reduced classes in* $\mathcal{C}_n^\mu$ *is* $\mathrm{LR} = \sum_{\mathcal{L}\in \mathrm{Im}(f)} \phi(n) \cdot n \cdot \mu!/|Is_s(\mathcal{L})|$.

5. *The number of set classes in* $\mathcal{C}_n^\mu$ *is* $\sum_{\mathcal{L}\in \mathrm{Im}(f)} \phi(n) \cdot n \cdot n!/|Is_s(\mathcal{L})| = \mathrm{LR} \cdot \frac{n!}{\mu!}$.

6. *The number of lists in* $\mathcal{C}_n^\mu$ *is* $\sum_{\mathcal{L}\in \mathrm{Im}(f)} \phi(n) \cdot n \cdot \mu! \cdot n!/|Is_s(\mathcal{L})| = \mathrm{LR} \cdot n!$.

Proof. A consequence of Lemma 4.4.1. □

4.6 Algorithms

There has been a history of errors in the enumeration of latin squares (this history is described in [69]). 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 arrived at the same results for $\mu \geq 2$ and $n \leq 14$. The results of this computation were independently verified by Fatih Demirkale. One of these algorithms when implemented 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 μ columns of size n that can generate lists of μ cyclic MNOLS of order n.

Algorithm A constructs all list-reduced lists of $\overline{\mu}$ 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 $(\overline{\mu} - 1)$ cyclic MNOLS of order n, and adding possible cyclic latin squares in order to yield lists of $\overline{\mu}$ cyclic MNOLS of order n that are list-reduced, where $2 \leq \overline{\mu} \leq \mu$. In the case the result is set-canonical, we calculate $|\mathrm{Red}_s(\mathcal{L})|$, $|\mathrm{Is}_l(\mathcal{L})|$, and $|\mathrm{Is}_s(\mathcal{L})|$. 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 4.5.1 to find the total number of classes.

Algorithm B works by using the set-canonical representation of a set-isotopy class of $(\overline{\mu}-1)$ cyclic MNOLS of order n, and adding possible cyclic latin squares in order to yield lists of $\overline{\mu}$ cyclic MNOLS of order n before checking whether the resulting lists are set-canonical, where $2 \leq \overline{\mu} \leq \mu$. If a result, say $\mathcal{L}$, is set-canonical, it is the canonical representation of a set-isotopy class of $\overline{\mu}$ cyclic MNOLS of order n. In such a case, we calculate $|\mathrm{Red}_s(\mathcal{L})|$, $|\mathrm{Is}_l(\mathcal{L})|$, and $|\mathrm{Is}_s(\mathcal{L})|$, before continuing to add more cyclic latin squares recursively. After completion, we merge the results and use Lemma 4.5.1 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 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 $A \in list1$, create a list $list3$ that contains each $B \in list2$ 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, $\mathcal{L}$, is set-canonical, we calculate $|\mathrm{Red}_s(\mathcal{L})|$, $|\mathrm{Is}_l(\mathcal{L})|$, and $|\mathrm{Is}_s(\mathcal{L})|$. After completion, we merge the results and use Lemma 4.5.1 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

Algorithm 1: Algorithm A

input : An integer $\overline{\mu}$ with $\overline{\mu} < \mu$;
A list of $\overline{\mu}$ columns $(C_1, \ldots, C_{\overline{\mu}})$ that generate a list of $\overline{\mu}$ 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 μ columns $(C_1, \ldots, C_{\overline{\mu}}, D_{\overline{\mu}+1}, \ldots, D_{\mu})$ that generate lists of μ cyclic MNOLS of order n that are (1) list-reduced, (2) set-reduced, (3) list-canonical, and (4) set-canonical;

sum $\leftarrow (0, 0, 0, 0)$;
function *extend1*$(\overline{\mu}, C_1, \ldots, C_{\overline{\mu}})$
function *extend1*$(\overline{\mu}, C_1, \ldots, C_{\overline{\mu}})$
for *columns P that form a permutation* **do**
  for $i = 1 : \overline{\mu}$ **do**
    if *The pair of columns (C_i, P) does not generate two cyclic MNOLS of order n* **then**
      go to the next possible column.
  $\mathcal{L} \leftarrow$ the list of μ cyclic MNOLS of order n generated by $(C_1, \ldots, C_\mu, P)$
  if $\overline{\mu} = \mu - 1$ **then**
    if $\mathcal{L}$ *is set-canonical* **then**
      sum $\leftarrow$ *sum* $+ (1, 1, 1, 1)$
    else
      if $\mathcal{L}$ *is list-canonical* **then**
        sum $\leftarrow$ *sum* $+ (1, 1, 1, 0)$
      else
        if $\mathcal{L}$ *is set-reduced* **then**
          sum $\leftarrow$ *sum* $+ (1, 1, 0, 0)$
        else
          sum $\leftarrow$ *sum* $+ (1, 0, 0, 0)$
  else
    sum $\leftarrow$ *sum* $+$ *extend*$1(\overline{\mu} + 1, C_1, \ldots, C_{\overline{\mu}}, P)$;
return sum;

Algorithm 2: Algorithm B

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

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

store $\leftarrow \emptyset$;
procedure *extend2*$(\overline{\mu}, C_1, \ldots, C_{\overline{\mu}})$
for *columns P that form a permutation* **do**
    for $i = 1 : \overline{\mu}$ **do**
        if *The pair of columns (C_i, P) does not generate a list of two cyclic MNOLS of order n* **then**
            break, and go to the next possible column P.
    $\mathcal{L} \leftarrow$ the list of $\overline{\mu} + 1$ cyclic MNOLS of order n generated by $(C_1, \ldots, C_{\overline{\mu}}, P)$
    if $\mathcal{L}$ *is set-canonical* **then**
        if $\overline{\mu} = \mu - 1$ **then**
            store $\leftarrow$ *store* $\cup \{(\mathrm{Is}_s(L), \mathrm{Is}_l(L), \mathrm{Red}_s(L))\}$
        else
            store $\leftarrow$ *store* $\cup$ *extend*$(\overline{\mu} + 1, C_1, \ldots, C_{\overline{\mu}}, P)$
return *store*

program as a whole.

Algorithm 3: Algorithm C

input : 1/ A list, $list1$, that contains all lists of columns such that $C \in list1$ 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 $C \in list2$ 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 μ cyclic MNOLS of order n with canonical representative $\mathcal{L}$, *store* contains one triple $(\mathrm{Is}_s(\mathcal{L}), \mathrm{Is}_l(\mathcal{L}), \mathrm{Red}_s(\mathcal{L}))$;

$store \leftarrow \emptyset$;
$\mathrm{vert}(graph) \leftarrow \emptyset$;
$\mathrm{edge}(graph) \leftarrow \emptyset$;
for $C_1 \in list1$ **do**
  $list3 \leftarrow \emptyset$
  for $C_2 \in list2$ **do**
    if (C_1, C_2) *generates a list of two cyclic MNOLS of order* n **then**
      $list3 \leftarrow list3 \cup C_2$
  $\mathrm{vert}(graph) \leftarrow list3$
  for $C_3 \in list3$ **do**
    for $C_4 \in list3$ **do**
      if (C_3, C_4) *generates a list of two cyclic MNOLS of order* n **then**
        $\mathrm{edge}(graph) \leftarrow \mathrm{edge}(graph) \cup \{C_3, C_4\}$
  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**
    $\mathcal{L} \leftarrow$ the list of μ cyclic MNOLS of order n generated by $(I, C_1, \alpha_1, \ldots, \alpha_{\mu-2})$
    $store \leftarrow store \cup \{(\mathrm{Is}_s(\mathcal{L}), \mathrm{Is}_l(\mathcal{L}), \mathrm{Red}_s(\mathcal{L}))\}$

4.7 Results and conclusions

The counts that were found appear in Tables 4.1, 4.2, 4.3, and 4.4. Comparing these results to the previously known cases in Table 1.1, 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 [61] that proposed the maximum μ for which a set of μ cyclic MNOLS of order n exists is $\lceil n/4 \rceil + 1$, 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 $n = 18$.

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 4.1: The number of two cyclic 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 4.2: The number of three cyclic MNOLS of order n under the given equivalence.

We say a list of μ cyclic MNOLS of order n is of type *type 0* if it is isotopically equivalent to a list of reduced μ cyclic MNOLS of order n, $\mathcal{L} = (L_1, \ldots, L_\mu)$, with $(0, 0, 1), (1, 0, 0) \in L_2$,

n	4	6	8	10	12	14	16
set-isotopy	0	0	0	1	2	117	14672
set-reduced	0	0	0	20	96	8638	1870592
list-isotopy	0	0	0	12	48	2484	350730
list-reduced	0	0	0	480	2304	207312	44879616

Table 4.3: The number of four cyclic 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

Table 4.4: The number of five cyclic MNOLS of order n under the given equivalence.

and is of *type 1* otherwise. A set of μ MNOLS of order n is of type 0 if fixing the order in some way gives a list of μ MNOLS of order n of type 0, and is of type 1 otherwise.

A collection of μ 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 μ cyclic 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 4.5, 4.6, 4.7, 4.8, 4.9, and 4.10 we show the number of set-isotopy classes of each type. Observe that the proportion of set-isotopy classes that are of type 0 increases as μ increases. This may be of interest in future searches for sets of μ cyclic MNOLS of order n where μ 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 $|\mathrm{Red}_s(\mathcal{L})| = 1$ for $n = 14$, so we omit the column for $|\mathrm{Red}_s(\mathcal{L})|$ in this case.

$\|\mathrm{Is}_s(\mathcal{L})\|$	$\|\mathrm{Is}_l(\mathcal{L})\|$	#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 4.5: The two cyclic MNOLS of order 14, by their type and autotopy group sizes.

$\|\mathrm{Is}_s(\mathcal{L})\|$	$\|\mathrm{Is}_l(\mathcal{L})\|$	#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 4.6: The three cyclic MNOLS of order 14, by their type and autotopy group sizes.

$\|\mathrm{Is}_s(\mathcal{L})\|$	$\|\mathrm{Is}_l(\mathcal{L})\|$	#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 4.7: The four cyclic MNOLS of order 14, by their type and autotopy group sizes.

$\lvert \mathrm{Is}_s(\mathcal{L}) \rvert$	$\lvert \mathrm{Is}_l(\mathcal{L}) \rvert$	$\lvert \mathrm{Red}_s(\mathcal{L}) \rvert$	#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 4.8: The two cyclic MNOLS of order 16, by their type and autotopy group sizes.

$\lvert \mathrm{Is}_s(\mathcal{L}) \rvert$	$\lvert \mathrm{Is}_l(\mathcal{L}) \rvert$	$\lvert \mathrm{Red}_s(\mathcal{L}) \rvert$	#Type 0	#Type 1	#Total
1	1	1	36845488	33760273	70605761
2	2	1	2326	666	2992
		total:	36847814	33760939	70608753

Table 4.9: The three cyclic MNOLS of order 16, by their type and autotopy group sizes.

$\lvert \mathrm{Is}_s(\mathcal{L}) \rvert$	$\lvert \mathrm{Is}_l(\mathcal{L}) \rvert$	$\lvert \mathrm{Red}_s(\mathcal{L}) \rvert$	#Type 0	#Type 1	#Total
1	1	1	11146	3401	14547
2	1	1	28	79	107
2	2	1	7	2	9
2	1	2	8	0	8
4	1	4	1	0	1
		total:	11190	3482	14672

Table 4.10: The four cyclic MNOLS of order 16, by their type and autotopy group sizes.

Chapter 5

Conclusion

This thesis has investigated a variety of concepts related to latin squares. In its chapters, we have established new results towards the existence and enumeration of three different structures, each related to latin squares.

In the second chapter we looked at the spectrum of μ transversals, of the back circulant latin square B_n, with common intersection. This included a number of basic constructions (subsection 2.2.1), a computer search when n was small (subsection 2.2.2), and a more substantial construction (subsection 2.2.3). This principal construction requires certain sets of μ partial transversals that occur in a given subsquare of B_n. In section 2.3, we investigated these partial transversals when $\mu = 3, 4$, providing a number of examples that were found with aid of a computer. This solved Question 1.3.1 for $\mu = 3, 4$ for a majority of cases, the results being summarized in Theorem 2.1.3 and Theorem 2.1.4. As an application of this work, we showed that μ transversals of B_n that intersect stably in t points could be used to construct $(\mu, n-t, n)$-latin trades (Theorem 2.4.2, for application to $\mu = 3, 4$ see Theorem 2.4.4 and Theorem 2.4.5).

There are three future directions that we envisage for this study. First, we could find the required sets of μ partial transversals for the principal construction when $\mu \geq 5$. For $\mu = 5, 6$, this might be achieved by a computer search. For $\mu \geq 7$, the amount of computation required appears to be unreasonably large, so it would be of interest to find a theoretical method to create the particular partial transversals required for the principal construction. Second, other constructions would need to be created, as the principal construction cannot construct examples when the intersection size of the transversals is very small or large, and when n is small. Third, it would be of interest to find when sets of μ transversals of common intersection do not exist, which we believe is what happens for the cases when the intersection of transversals is close to n.

In the third chapter, we investigated (μ, k, m)-latin trades. We used a variety of constructions to find idempotent (μ, k, m)-latin trades, namely:

- Basic constructions (Section 3.3)
- Computer searches (Subsection 3.4.1)
- Extended multiplication construction (Lemma 3.4.2)
- Packing construction (Theorem 3.4.3)(for $\mu = 3$, see Theorem 3.4.7)
- RPBD construction (Theorem 3.4.12) (for $\mu = 3$, see Lemma 3.4.18)

We combined these constructions in an inductive argument to yield $(3, k, m)$-latin trades for all but 196 exceptions (k, m), where $m \leq 194$. This solves Question 1.3.2 with $\mu = 3$ outside of these exceptions.

An obvious first direction for future work is to establish the existence of the 196 possible exceptions. Adding to this, solving Question 1.3.2 with $\mu = 4$ should be achievable using constructions presented in this thesis. For the cases with $\mu \geq 5$, future work will require greater effort. This is due to the fact that we need to know the spectrum of μ transversals in B_n with common intersection for Theorem 3.3.8, and we need to know the spectrum of volumes for μ-way latin trades of any fixed order for the RPBD construction (this has been done for $\mu = 4$ recently [4]).

In the fourth chapter, we investigated collections of mutually nearly orthogonal latin squares. The particular problem we dealt with was the enumeration of the number of collections of μ cyclic MNOLS of order n under a number of equivalences. We were able to construct 3 algorithms to solve this problem (see Section 4.6). The more advanced of these algorithms, Algorithm C, constructed representatives for each set-isotopy class of lists of μ MNOLS of order n, for $n \leq 16$ and $\mu \leq 5$. An integral part of this enumeration was knowing the form of any isotopism between μ cyclic MNOLS (Theorem 4.3.5) and having a canonical form, which we defined explicitly (Section 4.5). An amount of work went into optimizing the code for memory storage and running time, which was essential in order for the computation to complete in a reasonable amount of time.

There are a few directions for future work. For an immediate generalization, it seems it may be possible (with some difficulty) to solve the order 18 case, but the order 20 case seems impossible with present technology. A similar technique to ours was used to enumerate four MOLS of order 14 that can be created by difference matrices [81], so future work may also include similar investigations into MOLS. Alter [5] asked whether the number of reduced latin squares seemed to be divisible by an increasingly large number of twos, and further work has investigated this [16][72]. The number of set-isotopy classes of 3 cyclic MNOLS of order $12, 14, 16$ are respectively $2 \cdot 3 \cdot 733$, $3^3 \cdot 23 \cdot 691$, and $3^6 \cdot 96857$. There could perhaps be a similar phenomenon here with the divisibility by threes, so it may also be interesting to investigate divisibility properties of the number of cyclic MNOLS.

Bibliography

[1] P. Adams, E.J. Billington, D.E. Bryant, and Mahmoodian E.S. “The three-way intersection problem for Latin squares”. In: *Discrete Math.* 243.1-3 (2002), pp. 1–19 (cit. on pp. 8, 14, 58).

[2] P. Adams, E.J. Billington, D.E. Bryant, and A. Khodkar. “The μ-way intersection problem for m-cycle systems”. In: *Discrete Mathematics* 231.1-3 (2001), pp. 27–56 (cit. on p. 14).

[3] P. Adams, E.J. Billington, D.E. Bryant, and E.S. Mahmoodian. “On the possible volumes of μ-way Latin trades”. In: *Aequationes Math.* 63.3 (2002), pp. 303–320 (cit. on p. 14).

[4] P. Adams, E.S. Mahmoodian, H. Minooei, and M. Mohammadi Nevisi. “The four-way intersection problem for latin squares”. In: *arXiv:1408.6725* (2015) (cit. on pp. 8, 95).

[5] R. Alter. “How many Latin squares are there?” In: *Amer. Math. Monthly* 82 (1975), pp. 632–634 (cit. on p. 95).

[6] L.D. Andersen. *Chapter on The history of latin squares.* Research Report Series R-2007-32. Department of Mathematical Sciences, Aalborg University., 2007 (cit. on p. 1).

[7] B. Bagheri Gh., D.M. Donovan, and E.S. Mahmoodian. “On the existence of 3-way k-homogeneous Latin trades”. In: *Discrete Mathematics* 312.24 (2012), pp. 3473–3481 (cit. on pp. 10, 35, 38, 40–43, 46, 69).

[8] B. Bagheri Gh. and E.S. Mahmoodian. “On the existence of k-homogeneous Latin bitrades”. In: *Util. Math.* 85 (2011), pp. 333–345 (cit. on pp. 35, 41, 42).

[9] R. Bean. “Latin trades on three or four rows”. In: *Discrete Math.* 306.23 (2006), pp. 3028–3041 (cit. on p. 9).

[10] R. Bean, H. Bidkhori, M. Khosravi, and E.S. Mahmoodian. “k-homogeneous Latin trades”. In: *Bayreuth. Math. Schr.* 74 (2005), pp. 7–18 (cit. on pp. 35, 41, 42).

[11] E.J. Billington. “Combinatorial trades: a survey of recent results”. In: *Designs, 2002 Math. Appl., 563, Kluwer Acad. Publ., Boston, MA, 2003* (2003), pp. 47–67 (cit. on p. 35).

[12] R.C. Bose and S.S. Shrikhande. “On the construction of sets of mutually orthogonal Latin squares and the falsity of a conjecture of Euler”. In: *Trans. Amer. Math. Soc.* 95 (1960), pp. 191–209 (cit. on p. 2).

[13] R.C. Bose and S.S. Shrikhande. “On the falsity of Euler’s conjecture about the non-existence of two orthogonal Latin squares of order 4t+2”. In: *Proc. Nat. Acad. Sci. U.S.A.* 45 (1959), pp. 734–737 (cit. on p. 2).

[14] R.C. Bose, S.S. Shrikhande, and E.T. Parker. “Further results on the construction of mutually orthogonal Latin squares and the falsity of Euler’s conjecture.” In: *Canad. J. Math.* 12 (1960), pp. 189–203 (cit. on p. 2).

[15] J.W. Brown and E.T. Parker. “More on order 10 turn-squares”. In: *Ars Combin.* 35 (1993), pp. 125–127 (cit. on p. 6).

[16] J. Browning, D.S. Stones, and I.M. Wanless. “Bounds on the number of autotopisms and subsquares of a Latin square.” In: *Combinatorica* 33.1 (2013), pp. 11–22 (cit. on p. 95).

[17] J.M. Browning, P.J. Cameron, and I.M. Wanless. "Bounds on the number of small Latin subsquares". In: *J. Combin. Theory Ser. A* 124 (2014), pp. 41–56 (cit. on p. 17).

[18] P. Cameron. *Permutation Groups*. Cambridge University Press, 1999 (cit. on p. 80).

[19] N.J. Cavenagh. "A superlinear lower bound for the size of a critical set in a Latin square". In: *J. Combin. Des.* 15.4 (2007), pp. 269–282 (cit. on p. 9).

[20] N.J. Cavenagh. "A uniqueness result for 3-homogeneous Latin trades". In: *Comment. Math. Univ. Carolin.* 47.2 (2006), pp. 337–358 (cit. on pp. 10, 35, 41).

[21] N.J. Cavenagh. "Embedding 3-homogeneous Latin trades into abelian 2-groups". In: *Comment. Math. Univ. Carolin.* 45.2 (2004), pp. 191–212 (cit. on p. 10).

[22] N.J. Cavenagh. "Latin trade algorithms and the smallest critical set in a Latin square". In: *J. Autom. Lang. Comb.* 8.4 (2003), pp. 567–578 (cit. on p. 9).

[23] N.J. Cavenagh. "The theory and application of Latin bitrades: a survey". In: *Math. Slovaca* 58.6 (2008), pp. 691–718 (cit. on pp. 9, 35).

[24] N.J. Cavenagh, D.M. Donovan, and A. Drápal. "3-homogeneous Latin trades". In: *Discrete Math.* 300.1-3 (2005), pp. 57–70 (cit. on pp. 9, 35, 41, 42).

[25] N.J. Cavenagh, D.M. Donovan, and A. Drápal. "4-homogeneous Latin trades". In: *Australas. J. Combin.* 32 (2005), pp. 285–303 (cit. on pp. 35, 41, 42).

[26] N.J. Cavenagh, D.M. Donovan, and A. Khodkar. "On the spectrum of critical sets in back circulant Latin squares". In: *Ars Combin.* 82 (2007), pp. 287–319 (cit. on p. 9).

[27] N.J. Cavenagh, D.M. Donovan, and E. Şule Yazıcı. "Minimal homogeneous Latin trades". In: *Discrete Math.* 306.17 (2006), pp. 2047–2055 (cit. on p. 10).

[28] N.J. Cavenagh, C. Hämäläinen, and A.M. Nelson. "On completing three cyclically generated transversals to a latin square". In: *Finite Fields Appl.* 15.3 (2009), pp. 294–303 (cit. on p. 7).

[29] N.J. Cavenagh, J.S. Kuhl, and I.M. Wanless. "Longest partial transversals in plexes". In: *Ann. Comb.* 18.3 (2014), pp. 419–428 (cit. on p. 10).

[30] N.J. Cavenagh and I.M. Wanless. "On the number of transversals in Cayley tables of cyclic groups". In: *Discrete Appl. Math.* 158.2 (2010), pp. 136–146 (cit. on pp. 7, 14, 15, 41, 42, 46, 71).

[31] Y. Chang. "The existence of a large set of idempotent quasigroups of order 62". In: *J. Combin. Math. Combin. Comput.* 32 (2000), pp. 213–218 (cit. on p. 44).

[32] Y. Chang. "The spectrum for large sets of idempotent quasigroups". In: *J. Combin. Des.* 8.2 (2000), pp. 79–82 (cit. on p. 44).

[33] C. J. Colbourn and J. H. Dinitz, eds. *Handbook of Combinatorial Designs*. 2nd ed. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, 2007 (cit. on pp. 1, 6, 9, 11, 62).

[34] D. Curran and G.H.J. Van Rees. "Proceedings of the Eighth Manitoba Conference on Numerical Mathematics and Computing (Univ. Manitoba, Winnipeg, Man., 1978)". In: *Congress. Numer., XXII* (1979), pp. 165–168 (cit. on p. 9).

[35] F. Demirkale, D.M. Donovan, and A. Khodkar. "Direct constructions for general families of cyclic mutually nearly orthogonal Latin squares". In: *J. Combin. Des.* 23.5 (2015), pp. 195–203 (cit. on pp. 12, 74).

[36] F. Demirkale et al. "Difference Covering Arrays and Pseudo-Orthogonal Latin Squares". In: *Graphs and Combinatorics* (2015), pp. 1–22 (cit. on pp. 12, 74).

[37] D.M. Donovan and J. Cooper. "Critical sets in back circulant Latin squares". In: *Aequationes Math.* 52.1-2 (1996), pp. 157–179 (cit. on p. 7).

[38] D.M. Donovan, J.G. LeFevre, and G.H.J. van Rees. “On the spectrum of critical sets in Latin squares of order 2^n”. In: *J. Combin. Des.* 16.1 (2008), pp. 25–43 (cit. on p. 9).

[39] A. Drápal and T.S. Griggs. “Homogeneous toroidal Latin bitrades”. In: *Ars Combin.* 96 (2010), pp. 343–351 (cit. on p. 10).

[40] A. Drápal and T. Kepka. “Exchangeable partial groupoids I”. In: *Acta Univ. Carolin. Math. Phys.* 24.2 (1983), pp. 57–72 (cit. on p. 8).

[41] P.J. Dukes and J. Howell. “Solution of the intersection problem for Latin squares of different orders”. In: *J. Combin. Math. Combin. Comput.* 80 (2012), pp. 289–298 (cit. on p. 8).

[42] J. Egan and I.M. Wanless. “Enumeration of MOLS of small order”. In: *Math. Comp.* 85.298 (2016), pp. 799–824 (cit. on p. 73).

[43] L. Euler. *Investigations on a new type of magic square.* Translated by Ho, A. and Klyve, D. (cit. on p. 1).

[44] L. Euler. “Recherches sur une nouvelle espece de quarres magiques”. In: *Verhandelingen uitgegeven door het zeeuwsch Genootschap der Wetenschappen te Vlissingen* 9 (1782), pp. 85–239 (cit. on p. 1).

[45] I.A. Faradžev. “Generation of nonisomorphic graphs with a given distribution of the degrees of vertices. (Russian)”. In: *Algorithmic studies in combinatorics (Russian)* 185 (1978), pp. 11–19 (cit. on p. 73).

[46] C.M. Fu and H.-L. Fu. “The intersection of three distinct Latin squares”. In: *Matematiche (Catania)* 44.1 (1989), pp. 21–45 (cit. on p. 8).

[47] C.M. Fu, H.-L Fu, and S.H. Guo. “The intersections of commutative Latin squares”. In: *Ars Combin.* 32 (1991), pp. 77–96 (cit. on p. 8).

[48] C.M. Fu, Y.H Gwo, and F.C. Wu. "The intersection problem for semi-symmetric Latin squares." In: *J. Combin. Math. Combin. Comput.* 23 (1997), pp. 47–63 (cit. on p. 8).

[49] C.M. Fu, W.C. Huang, Y.H. Shih, and Y.J. Yaon. "Totally symmetric Latin squares with prescribed intersection numbers". In: *Discrete Math.* 282.1-3 (2004), pp. 123–136 (cit. on p. 8).

[50] H.-L. Fu. "More results on the intersections of Latin squares". In: *J. Inform. Optim. Sci.* 11.3 (1990), pp. 525–535 (cit. on p. 8).

[51] H.-L. Fu. "On the construction of certain trypes of latin squares with prescribed intersections". PhD thesis. Auburn university, 1980 (cit. on p. 8).

[52] M. J. Grannell, T. S. Griggs, and M. Knor. "Biembeddings of symmetric configurations and 3-homogeneous Latin trades". In: *Comment. Math. Univ. Carolin.* 49.3 (2008), pp. 411–420 (cit. on p. 41).

[53] C. Hämäläinen. "Partitioning 3-homogeneous Latin bitrades". In: *Geom. Dedicata* 133 (2008), pp. 181–193 (cit. on p. 41).

[54] A. Hulpke, P. Kaski, and P.R.J. Östergård. "The number of Latin squares of order 11". In: *Math. Comp.* 80.274 (2011), pp. 1197–1219 (cit. on p. 82).

[55] P. Kaski and P.R.J. Östergård. *Classification algorithms for codes and designs.* Algorithms Comput. Math. 15, Springer, 2006 (cit. on p. 73).

[56] A.D. Keedwell. "Critical sets in latin squares and related matters: an update". In: *Util. Math.* 65 (2004), pp. 97–131 (cit. on p. 9).

[57] G. Kolesova, C.W.H. Lam, and L. Thiel. "On the number of 8×8 Latin squares". In: *J. Combin. Theory Ser. A* 54.1 (1990), pp. 143–148 (cit. on p. 73).

[58] C. Laywine and G. Mullen. *Discrete mathematics using latin squares.* Wiley-Interscience Series in Discrete Mathematics and Optimization, 1998 (cit. on p. 72).

[59] J.G. Lefevre, D.M. Donovan, and A. Drápal. "Permutation representation of 3 and 4-homogeneous Latin bitrades". In: *Fund. Inform.* 84.1 (2008), pp. 99–110 (cit. on pp. 35, 41).

[60] J.G. Lefevre and T.A. McCourt. "The disjoint m-flower intersection problem for Latin squares". In: *Electron. J. Combin.* 18.1 (2011), Paper 42, 33 pp (cit. on p. 8).

[61] P.C. Li and G.H.J van Rees. "Nearly orthogonal Latin squares". In: *J. Combin. Math. Combin. Comput.* 62 (2007), pp. 13–24 (cit. on pp. 11–13, 74, 79, 89).

[62] T.G. Marbach. 3-*Way k-homogeneous Latin trades - Programs A and B*. URL: https://github.com/triliu/3WaykHomogeneousLatinTrades (cit. on p. 64).

[63] T.G. Marbach. "On the intersection of three or four transversals of the back circulant latin square B_n". In: *Australas. J. Combin.* 65.1 (2016), pp. 84–107 (cit. on p. 7).

[64] T.G. Marbach and L. Ji. "The spectrum for 3-way k-homogeneous Latin trades". In: *Electron. J. Combin.* 22.4 (2015), Paper 4.1, 23pp (cit. on p. 10).

[65] T.M. Marbach, F. Demirkale, and D.M. Donovan. "The Enumeration of Cyclic MNOLS". In: *arXiv:1512.00997* (2015) (cit. on p. 13).

[66] T.A. McCourt. "On defining sets in Latin squares and two intersection problems, one for Latin squares and one for Steiner triple systems". In: *Bull. Aust. Math. Soc.* 82.2 (2010), pp. 351–352 (cit. on p. 8).

[67] B.D. McKay. "Isomorph-free exhaustive generation". In: *J. Algorithms* 26.2 (1998), pp. 306–324 (cit. on p. 73).

[68] B.D. McKay, J.C. McLeod, and I.M. Wanless. "The number of transversals in a latin square". In: *Des. Codes Cryptogr.* 40 (2006), pp. 269–284 (cit. on p. 6).

[69] B.D. McKay, A. Meynert, and W.J. Myrvold. "Small Latin Squares, Quasigroups and Loops". In: *J. Combin. Des.* 15.2 (2007), pp. 98–119 (cit. on pp. 6, 84).

[70] B.D. McKay and A. Piperno. “Practical Graph Isomorphism, II”. In: *J. Symbolic Comput.* 60 (2014), pp. 94–112 (cit. on p. 82).

[71] B.D. McKay and I.M. Wanless. “Most Latin squares have many subsquares”. In: *J. Combin. Theory Ser. A* 86.2 (1999), pp. 322–347 (cit. on p. 73).

[72] B.D. McKay and I.M. Wanless. “On the number of Latin squares”. In: *Ann. Comb.* 9.3 (2005), pp. 335–344 (cit. on pp. 73, 82, 95).

[73] S. Milici and G. Quattrocchi. “On the intersection problem for three Steiner triple systems.” In: *Ars Combin.* 24.A (1987), pp. 175–194 (cit. on p. 14).

[74] E.B. Pasles. “Mutually nearly orthogonal latin squares and their applications”. PhD thesis. Temple University, 2004 (cit. on pp. 11, 12).

[75] E.B. Pasles and D. Raghavarao. “Mutually nearly orthogonal Latin squares of order 6”. In: *Util. Math.* 65 (2004), pp. 65–72 (cit. on pp. 12, 73).

[76] D. Raghavarao, S.S. Shrikhande, and M.S. Shrikhande. “Incidence matrices and inequalities for combinatorial designs”. In: *J. Combin. Des.* 10.1 (2002), pp. 17–26 (cit. on pp. 11, 12, 72, 73).

[77] R.C. Read. “Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations.” In: *Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976). Ann. Discrete Math.* 2 (1978), pp. 107–120 (cit. on p. 73).

[78] G.H.J. van Rees. “Private communication”. 2015 (cit. on p. 74).

[79] B. Smetaniuk. “On the minimal critical set of a Latin square”. In: *Utilitas Math.* 16 (1979), pp. 97–100 (cit. on p. 9).

[80] L. Teirlinck and C.C. Lindner. “The construction of large sets of idempotent quasigroups”. In: *European J. Combin.* 9.1 (1988), pp. 83–89 (cit. on p. 44).

[81] D.T. Todorov. "Four Mutually Orthogonal Latin squares of order 14". In: *J. Combin. Des.* 20.8 (2012), pp. 363–367 (cit. on pp. 3, 95).

[82] W.D. Wallis. *Combinatorial Designs*. Monographs and Textbooks in Pure and Applied Mathematics, 1988 (cit. on p. 59).

[83] I.M. Wanless. "A generalisation of transversals for Latin squares". In: *Electron. J. Combin.* 9.1 (2002), R12 (cit. on p. 6).

[84] I.M. Wanless. "Diagonally cyclic Latin squares". In: *European J. Combin.* 25.3 (2004), pp. 393–413 (cit. on p. 7).

[85] I.M. Wanless. "Transversals in Latin squares". In: *Quasigroups Related Systems* 15.1 (2007), pp. 169–190 (cit. on p. 7).

[86] I.M. Wanless. "Transversals in Latin squares: a survey. Surveys in combinatorics 2011". In: *London Math. Soc. Lecture Note Ser.* 392 (2011), pp. 403–437 (cit. on p. 14).

[87] I.M. Wanless and B.S. Webb. "The existence of latin squares without orthogonal mates". In: *Des. Codes Cryptog.* 40.1 (2006), pp. 131–135 (cit. on p. 6).

Appendix A

Appendix

A.1 Base blocks for Chapter 2

$b+d$	intersect	result
9	0	8 6 7 2 0 1 5 3 4
		7 8 6 1 2 0 4 5 3
		5 3 4 8 6 7 2 0 1
		4 5 3 7 8 6 1 2 0
9	1	7 8 6 2 0 1 4 5 3
		8 5 3 7 0 6 1 2 4
		6 4 5 8 0 7 3 1 2
		5 7 8 3 0 2 6 4 1
10	0	8 9 5 2 0 1 6 7 3 4
		7 5 8 1 9 0 2 4 6 3
		6 4 2 9 7 3 8 0 5 1
		5 3 4 7 8 9 1 6 0 2
11	0	10 7 5 2 0 1 9 6 8 3 4
		7 9 6 1 2 0 10 8 4 5 3
		8 4 2 10 3 9 0 5 7 1 6
		6 3 10 7 1 2 8 9 0 4 5

Table A.1: $\mu = 4$ and $b = 9$.

$b+d$	intersect	result
12	0	10 7 5 2 0 1 9 11 8 3 4 6
		7 11 6 1 2 0 10 3 9 5 8 4
		8 9 2 4 1 11 0 5 10 6 7 3
		5 3 4 11 6 10 1 9 0 8 2 7
13	0	12 7 5 2 0 1 11 3 8 10 4 9 6
		8 6 12 1 2 0 4 11 9 3 10 5 7
		10 11 2 4 1 3 0 7 12 9 6 8 5
		5 3 4 11 12 7 2 0 10 1 9 6 8
14	0	12 7 5 2 0 1 13 3 8 11 4 10 6 9
		8 6 12 1 2 0 4 11 13 3 9 5 10 7
		10 13 2 4 1 3 0 5 12 9 6 11 7 8
		5 3 4 7 12 9 2 0 10 13 11 1 8 6
15	0	14 7 5 2 0 1 12 3 4 13 6 10 8 11 9
		8 6 14 1 2 0 4 7 13 3 12 9 11 5 10
		12 9 2 4 1 3 0 13 8 5 14 11 6 10 7
		5 3 4 9 10 11 2 0 14 1 13 7 12 8 6
16	0	14 7 5 2 0 1 12 3 4 15 6 11 8 13 9 10
		8 6 14 1 2 0 4 7 15 3 12 9 13 5 10 11
		12 9 2 4 1 3 0 13 8 5 14 15 6 10 11 7
		5 3 4 11 12 7 2 0 10 1 15 13 14 9 6 8

Table A.2: $\mu = 4$ and $b = 9$.

$b+d$	intersect	result
17	0	16 7 5 2 0 1 8 3 4 15 12 9 6 13 14 10 11
		8 6 16 1 2 0 4 7 14 3 10 5 15 11 12 13 9
		14 9 2 4 1 3 0 5 16 13 6 7 8 15 11 12 10
		5 3 4 15 10 7 2 0 8 1 14 16 13 9 6 11 12

Table A.3: $\mu = 4$ and $b = 9$.

$b+d$	intersect	result
9	2	8 6 7 2 0 1 5 3 4
		7 5 8 2 0 3 6 4 1
		6 8 5 2 0 7 4 1 3
9	3	7 5 8 2 0 3 6 4 1
		6 3 8 2 4 7 5 0 1
		4 7 8 2 3 6 0 5 1

Table A.4: $\mu = 3$ and $b = 9$.

$b+d$	intersect	result
11	0	10 8 9 5 2 0 1 6 7 3 4
		9 10 6 7 1 2 0 8 4 5 3
		8 9 10 2 7 1 3 0 6 4 5
		7 5 3 10 8 9 4 2 0 6 1
11	1	10 8 9 5 2 0 1 6 7 3 4
		9 7 5 3 10 0 6 8 2 4 1
		8 9 4 10 3 0 5 7 1 6 2
		7 5 10 8 4 0 9 2 6 1 3
11	2	10 8 9 5 2 0 1 6 7 3 4
		9 7 10 4 2 0 5 8 6 1 3
		8 9 5 10 2 0 3 7 4 6 1
		7 10 6 9 2 0 8 3 1 4 5
11	3	8 9 10 4 2 0 5 7 1 6 3
		8 10 5 2 9 7 0 3 1 6 4
		8 5 3 10 7 9 4 0 1 6 2
		8 4 9 3 10 2 7 5 1 6 0

Table A.5: $\mu = 4$ and $b = 11$.

$b+d$	intersect	result
12	3	9 11 6 7 2 0 1 10 8 5 3 4
		8 5 7 2 9 11 1 10 6 3 0 4
		6 9 3 11 5 8 1 10 0 7 2 4
		5 7 11 3 8 9 1 10 2 0 6 4
13	3	12 9 7 5 2 0 1 10 11 6 8 3 4
		7 9 6 3 1 12 10 2 11 5 8 4 0
		6 9 12 2 4 10 3 0 11 7 8 1 5
		5 9 10 4 12 1 2 7 11 0 8 6 3
14	3	12 13 7 5 2 0 1 3 11 9 10 6 4 8
		10 13 6 4 2 0 11 12 1 3 8 5 9 7
		8 13 5 9 2 0 3 11 12 1 6 10 7 4
		7 13 8 6 2 0 12 1 4 10 11 9 5 3
15	3	14 11 7 5 2 0 1 3 12 13 8 10 4 6 9
		8 9 14 6 2 0 1 13 11 3 4 12 10 5 7
		10 8 6 11 2 0 1 5 14 12 13 9 7 3 4
		9 7 8 13 2 0 1 14 4 11 12 3 6 10 5
16	3	14 15 7 5 2 0 1 3 4 13 11 9 6 12 10 8
		12 8 6 7 2 0 1 15 13 14 10 3 4 5 11 9
		9 7 8 13 2 0 1 14 6 3 15 12 10 11 4 5
		8 9 14 6 2 0 1 5 12 15 4 11 13 10 7 3

Table A.6: $\mu = 4$ and $b = 11$.

$b+d$	intersect	result
17	3	16 13 7 5 2 0 1 3 4 14 15 9 6 11 8 12 10
		12 8 6 7 2 0 1 16 14 15 10 3 4 5 13 11 9
		9 7 8 15 2 0 1 13 16 3 4 5 11 14 12 10 6
		8 9 12 6 2 0 1 5 15 7 14 16 10 13 11 3 4
18	3	16 13 7 5 2 0 1 3 4 17 15 9 6 11 8 14 12 10
		12 8 6 7 2 0 1 17 14 16 10 3 4 5 15 13 11 9
		9 7 8 17 2 0 1 11 16 3 4 5 15 13 14 10 6 12
		8 9 12 6 2 0 1 5 10 15 16 14 17 3 13 7 4 11
19	3	18 13 7 5 2 0 1 3 4 17 6 16 8 9 10 14 15 11 12
		12 8 6 7 2 0 1 15 16 18 10 3 4 5 17 13 14 9 11
		9 7 8 17 2 0 1 11 18 3 4 5 12 16 13 15 6 14 10
		8 9 12 6 2 0 1 5 14 15 18 7 17 3 16 11 4 10 13
20	3	18 13 7 5 2 0 1 3 4 19 6 15 8 9 10 17 14 16 11 12
		12 8 6 7 2 0 1 19 14 11 18 3 4 5 17 9 16 13 15 10
		9 7 8 17 2 0 1 15 16 3 4 5 6 19 12 18 13 14 10 11
		8 9 12 6 2 0 1 5 18 15 10 7 19 3 16 14 17 11 4 13
21	3	20 13 7 5 2 0 1 3 4 9 6 19 16 11 8 18 10 17 14 12 15
		12 8 6 7 2 0 1 19 20 13 10 3 4 5 18 9 15 16 17 11 14
		9 7 8 19 2 0 1 17 12 3 4 5 6 18 20 11 14 15 10 16 13
		8 9 12 6 2 0 1 5 14 11 18 15 19 3 4 20 17 7 16 13 10

Table A.7: $\mu = 4$ and $b = 11$.

$b+d$	intersect	result
11	4	9 10 8 5 2 0 1 6 7 3 4
		10 6 7 8 2 0 9 1 5 3 4
		7 9 6 10 2 0 5 8 1 3 4
11	5	10 8 6 9 2 0 1 4 7 5 3
		10 7 9 4 2 0 6 8 1 5 3
		10 6 7 8 2 0 9 1 4 5 3

Table A.8: $\mu = 3$ and $b = 11$.

$b+d$	intersect	result
15	1	14 12 13 9 7 5 2 0 1 10 11 6 8 3 4
		13 14 12 8 5 3 11 0 10 1 2 9 4 6 7
		12 13 11 14 4 10 3 0 2 9 1 5 7 8 6
		11 9 14 5 8 4 13 0 6 12 10 7 1 2 3
15	2	14 12 13 9 7 5 2 0 1 10 11 6 8 3 4
		13 14 10 11 12 4 2 0 3 1 9 7 5 8 6
		12 13 14 7 5 8 2 0 10 11 1 4 9 6 3
		11 9 12 6 14 10 2 0 13 8 3 1 4 7 5
15	3	14 12 13 9 7 5 2 0 1 10 11 6 8 3 4
		13 14 12 8 6 7 2 0 1 11 9 10 4 5 3
		11 13 10 7 14 12 2 0 1 4 6 9 3 8 5
		12 10 8 13 11 14 2 0 1 5 3 7 9 4 6
15	4	14 12 13 9 7 5 2 0 1 10 11 6 8 3 4
		13 11 14 8 10 5 2 0 1 12 9 7 3 4 6
		12 14 9 10 13 5 2 0 1 11 8 3 4 6 7
		11 13 10 12 14 5 2 0 1 4 6 9 7 8 3
15	5	14 12 13 9 7 5 2 0 1 10 11 6 8 3 4
		13 11 14 8 10 5 2 0 1 12 7 9 3 6 4
		12 14 9 10 13 5 2 0 1 11 6 3 7 8 4
		11 13 10 14 12 5 2 0 1 6 3 8 9 7 4

Table A.9: $\mu = 4$ and $b = 15$.

$b+d$	intersect	result
15	15	14 12 13 9 7 5 2 0 1 10 11 6 8 3 4
16	16	14 12 15 13 7 5 2 0 1 3 11 9 10 6 4 8
17	17	14 15 16 10 7 5 2 0 1 3 13 11 12 8 6 4 9
18	18	16 17 12 10 7 5 2 0 1 3 15 13 11 14 8 6 4 9
19	19	16 17 18 10 7 5 2 0 1 3 ... 4 15 13 11 14 8 6 12 9
20	20	18 19 14 10 7 5 2 0 1 3 ... 4 17 15 16 12 9 6 8 13 11
21	21	20 17 14 10 7 5 2 0 1 3 4 ... 18 19 15 13 9 6 16 8 11 12
22	22	20 21 14 10 7 5 2 0 1 3 4 ... 19 6 16 18 13 8 9 17 15 11 12
23	23	22 19 14 10 7 5 2 0 1 3 4 21 ... 6 20 17 13 8 9 18 11 15 16 12
24	24	22 23 14 10 7 5 2 0 1 3 4 9 6 ... 21 19 15 20 11 8 18 12 17 13 16
25	25	24 21 14 10 7 5 2 0 1 3 4 9 6 22 ... 23 15 20 11 8 19 12 13 18 16 17
26	26	24 25 14 10 7 5 2 0 1 3 4 9 6 23 8 ... 22 18 11 12 13 20 21 17 15 19 16
27	27	26 23 14 10 7 5 2 0 1 3 4 9 6 25 8 ... 24 18 11 12 13 22 15 19 21 16 20 17
28	28	26 27 14 10 7 5 2 0 1 3 4 9 6 11 8 25 ... 22 17 12 13 24 15 23 20 18 16 21 19
29	29	28 25 14 10 7 5 2 0 1 3 4 9 6 11 8 27 ... 24 17 12 13 26 15 16 22 20 18 23 21 19

Table A.10: $\mu = 4$ and $b = 15$ (Only one partial transversal is needed in each case).

A.2 Program details

A.2.1 More details

In this appendix, we will give some further details of our implementation of Algorithm C from Chapter 4.

Permutation objects and representing cyclic MNOLS

A list of cyclic $MNOLS$ can be represented by the first column of each of the contained latin squares:

$$\left\{ \begin{array}{|c|c|c|c|} \hline 0 & 1 & 2 & 3 \\ \hline 1 & 2 & 3 & 0 \\ \hline 2 & 3 & 0 & 1 \\ \hline 3 & 0 & 1 & 2 \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline 1 & 2 & 3 & 0 \\ \hline 3 & 0 & 1 & 2 \\ \hline 0 & 1 & 2 & 3 \\ \hline 2 & 3 & 0 & 1 \\ \hline \end{array} \right\} \rightarrow \begin{array}{|c|c|} \hline 0 & 1 \\ \hline 1 & 3 \\ \hline 2 & 0 \\ \hline 3 & 2 \\ \hline \end{array}$$

In our program, we will often store a list of cyclic $MNOLS$ in this array notation. Any column which may be a column in this array notation must be a permutation. As each column in this array notation is a permutation, we will often use a permutation object to hold these potential columns. We will say that two permutations are nearly orthogonal if the corresponding latin squares are nearly orthogonal.

These permutation objects are initiated to hold the identity permutation. One of the key functions on the permutation object is the `next()` function. The `next()` function modifies the current permutation in the permutation object to be the next lexicographically higher permutation.

For example, say the permutation object `Perm` contained the permutation $(0, 1, 2, 3, 4, 5)$. After calling `Perm.next()`, `Perm` will contain the permutation $(0, 1, 2, 3, 5, 4)$, and after calling it a second time `Perm` will contain the permutation $(0, 1, 2, 5, 3, 4)$.

Rather than stepping through every permutation like this, it will be possible to skip over a great number of them. Suppose the permutation object `Perm` contains the permutation $(2,1,3,4,5,0)$. This permutation is not nearly orthogonal to the identity permutation $(0,1,2,3,4,5)$ because both contain the symbol 1 in the second entry. If we wish to find the next permutation that is nearly orthogonal to the identity permutation, clearly the next few permutations of $(2,1,3,5,0,4)$, $(2,1,3,5,4,0)$, etc. will each have the same problem of containing the symbol 1 in the second entry. For this, we use the function `Perm.next(incorrect_digit)`, which modifies the current permutation in the permutation object to be the next lexicographically higher permutation that no longer contains an incorrect digit at index `incorrect_digit`. In this case, the result of the function call `Perm.next(1)` would mean `Perm` would contain the permutation $(2,3,0,1,4,5)$.

CheckMNOLS

This function was run many times, so we spent some effort to optimize it. The fastest method we could find is the one presented here. Given two permutations Q and P, we wish to return whether the two are nearly orthogonal or not. Recall that this means each difference $P(i) - Q(i)$ occurs exactly once, except when the difference is 0, where no such differences occur, and when the difference is $n/2$, where there are two such occurrences. We constructed an array `temp_all` (which is declared externally). The purpose of this array is to hold the number of occurrences of each difference, although we modify this slightly by initializing `temp_all[0]=1` and `temp_all[n/2]=-1`. Whenever a difference of `diff` is found, we increase `temp_all[diff]` by 1. Thus, if Q and P are nearly orthogonal, after this process `temp_all[i]=1` for $0 \leq i \leq n-1$. Programming in this way simplifies the branching conditions of the function.

Check_Canonical and Count_Autotopisms

Our rudimentary method for both these functions cycle through every group action of $\langle M, T, R, S\rangle$ and apply it to the incoming list of $MNOLS$, Z. For `Check_Canonical`, we ensured each of these was larger than or equal to Z. For `Count_Autotopisms`, we counted how many group actions (of each of the groups we are concerned with) returned Z.

OpenMP

We used the API *openMP* to split the computation into multiple threads. The function `omp_set_num_threads` determines how many threads our program will be split into. We found that n threads worked more efficiently that using more or less threads.

When programming multithreaded applications, we have to be careful of race conditions. This is when two threads simultaneously read or write at the same memory address, causing the program to have a bug. To this end, we use atomics and locks. The line `#pragma omp atomic` causes the next line to be done in serial. We have used this for most memory stores that are not local. The functions `omp_set_lock` and `omp_unset_lock` respectively set and unset a variable. If the variable has been locked by a thread, another thread cannot enter that segment in the code until the original thread has unlocked the variable. These are used when we wish to lock several lines of code, rather than just changing one variable.

Unraveled cliques

This program in general required us to solve the clique problem. However, as we were searching for five $MNOLS$ of order 16, the corresponds clique would be a clique of size 3, which is small. In the code presented here, we have unraveled the clique problem. On finding an edge of our graph, rather than adding that edge to a graph object, we have looped through the other possible vertices and tried to find a third vertex to form a clique of size 3. Although this slows down our code slightly, this slowdown in negligible. This simplifies our code and gives us more confidence that it is error free.

A.2.2 Code

```
#include <stdio.h>
#include <tchar.h>

#include <omp.h>
#include <ctime>

#include <fstream>
#include<string>

#include <list>
#include <algorithm>
#include <vector>
#include <stdio.h>
#include <iostream>
using namespace std;


//Input: two integers
//Output: the greatest integer that divides both input integers
int GCD(int a, int b) {
  if (a == 0) {
    return b;
  }
  if (b == 0) {
    return a;
  }
  if (a <= b) {
    if (b%a == 0) {
      return a;
    }
    return GCD(b - (b / a)*a, a);
  }
  else {
    return GCD(b, a);
  }
}

//Input: integer 'n'
//Output: n!
int factorial(int n) {
  if (n < 1) { return 0; }
  int temp = 1;
  for (int i = 2; i <= n; i++) {
    temp *= i;
  }
  return temp;
}

//Input: integer 'n'
```

```
//Output: Eulers totient function of 'n' (i.e. the number of invertible
    elements in mod n)
int totient(int n) {
  int temp = 0;
  for (int i = 1; i < n; i++) {
    if (GCD(i, n) == 1) {
      temp++;
    }
  }
  return temp;
}
```

```
//Creates an object that imitates a permutation
class Permutation
{
public:
  int get(int i);
  void set(int Place_i, int value);
  Permutation() {};

  //Input: size 'k'
  //Creates a permutation object of size 'k'
  Permutation(int k) {
    //on initialization, we set this permutation object to be the identity
        permutation
    not_Finished = true;
    Store = new int[k];
    kk = k;
    for (int i = 0; i < k; i++) {
      Store[i] = i;
    };
  }
  int size_k() { return kk; }
  void next();
  void next(int);
  void reset();
  void reset(int);
  bool notFinished() { return not_Finished; }
  void setFinished() { not_Finished = false; }
  int reverseGet(int i);
  void del();
  void Print() {
    for (int i = 0; i < kk; i++) {
      cout << Store[i] << ' ';
    }
    cout << '\n';
  }
private:
  bool not_Finished;
  int *Store;
  int kk;
};

//Input: a symbol 'i'
//Output: the index of the current permutation that containing symbol 'i'
int Permutation::reverseGet(int i) {
  for (int j = 0; j < kk; j++) {
    if (Store[j] == i) {
```

```
      return j;
    }
  }
  return -1;
}

//Input: an index 'i'
//Output: the symbol of the current permutation at index i
int Permutation::get(int i)
{
  int d;
  d = Store[i];
  return d;
}

//Caution: Dangerous function
//Input: an index 'Place_i' and a symbol 'value'
//Result: this permutation object changes the value in index 'Place_i' to be
    value 'value'
void Permutation::set(int Place_i, int value)
{
  Store[Place_i] = value;
}


//Result: this permutation object is set to contain identity permutation
    (0,1,....,kk-1)
void Permutation::reset() {
  for (int i = 0; i < kk; i++) { Store[i] = i; };
  not_Finished = true;
}

//Input: an index 'from_position'
//Result: those entries of perm from index 'from_position' to index 'kk'-1 are
    placed in accending order
void Permutation::reset(int from_position) {
  sort(Store + from_position, Store + kk);
}

//Input: an index 'incorrect_digit'
//Result: this permutation object is set to be the next highest (w.r.t
    lexicographic ordering) possible
//          permutation such that the value in index 'incorrect_digit' has
    changed
void Permutation::next(int incorrect_digit) {

  //In this case, we have run out of permutations
  if (incorrect_digit<0) {
    not_Finished = false;
    return;
  }
  //Swaps perm(incorrect digit_ with the next largest perm(i) with i>
      incorrect_digit,
  //  and sorts all the perm(i') with i'>incorrect_digit into natural order.
  int smallest_greaterthen_id = kk;
  for (int i = incorrect_digit + 1; i<kk; i++) {
    if (Store[i] < smallest_greaterthen_id && Store[i]>Store[incorrect_digit])
        {
      smallest_greaterthen_id = Store[i];
```

```
    }
  }
  //If none to swap with, try increasing from a smaller index
  if (smallest_greaterthen_id == kk) {
    next(incorrect_digit - 1);
    return;
  }
  else {
    for (int i = incorrect_digit + 1; i<kk; i++) {
      if (Store[i] == smallest_greaterthen_id) {
        //Perform swap
        Store[i] = Store[incorrect_digit];
        Store[incorrect_digit] = smallest_greaterthen_id;
        reset(incorrect_digit + 1);
        break;
      }

    }
  }
}

//Result: this permutation object is set to be the next highest (w.r.t
    lexicographic ordering) possible permutation
void Permutation::next()
{
  next(kk - 1);
}

//Result: delete this permutation object's free store data
void Permutation::del() {
  delete[] Store;
}

//Input: two integers 'n','r'
//Output: the number of r-subsets in an n-set
int nCr(int n, int r) {
  int prod = 1;
  //int temp = n;
  for (int i = 0; i < r; i++) {
    prod *= (n - i);
  }
  for (int i = 1; i <= r; i++) {
    prod /= i;
  }
  return prod;
}

//Input: two permutations 'Q','P' and a set of integers 'temp_all'
//Output: true, if the permutations 'Q','P' are nearly orthogonal
//        false, otherwise.
//Note: temp_all is an input to save on allocation time of declaring locally
bool checkMNOLS(Permutation &Q, Permutation &P, unsigned __int64* temp_all) {
```

```
  int n = P.size_k();

  //temp_all is a device that holds the number of possible differences
  //when filling in temp_all with differences, it will be convinient to fail
      if the number of differences exceeds 1
  //  if the difference is neither 0 nor n/2, then we cannot have more or less
      than 1 occurance of that difference
  //  if the difference is 0, then we should fail. Set temp_all[0] = 1 at
      initiation, so any occurance of difference 0 causes a fail
  //  if the difference is n/2, then we should only fail after 2 occurance.
      Set temp_all[0] = -1 at initiation, so exactly two occurance of
      difference n/2 will not cause a fail

  //reset 'temp_all'
  for (int i = 0; i < n; i++) {
    temp_all[i] = 0;
  }
  //we can have two symbols that are n/2
  temp_all[n / 2] = -1;
  //we can have zero symbols that are 0
  temp_all[0] = 1;


  int diff;
  int Q_i;
  int P_i;
  int numb_this_diff;

  //for each index i, calculate the difference of the two permutations at
      index i,
  //  if this difference has temp_all[difference] <1, increase
      temp_all[difference]
  //  otherwise, these permutations are not nearly orthogonal.
  //            -Update permutation object P to next permutation without a
      similar issue to this permutation
  //            -Return false for this entire checkMNOLS function.
  for (int i = 0; i < n; i++) {
    Q_i = Q.get(i);
    P_i = P.get(i);
    diff = (P_i - Q_i < 0) ? P_i - Q_i + n : P_i - Q_i;
    numb_this_diff = temp_all[diff];
    //each difference can occur once, exept for a difference of 0 (which never
        occurs) and a difference of n/2 (which occurs twice)
    if (numb_this_diff == 1) {
      //this is a fail at position i
      P.next(i);
      return false;
    }
    else {
      //increase numb_this_diff by one
      temp_all[diff]++;
    }
  }

  //If no problems were found, these permutations are nearly orthogonal, so
      return true.
  return true;
}



//Input: two arrays of integers 'smaller','larger'
```

```
//Output: returns true if 'smaller' is less than 'larger' under the total
    ordering <| defined in the chapter
bool lower(unsigned __int64** smaller, unsigned __int64** larger, int n, int
    k) {

  for (int i = 0; i<k; i++) {
    for (int j = 0; j < n; j++) {
      if (smaller[i][j] < larger[i][j]) {
        return true;
      }
      else if (smaller[i][j] > larger[i][j]) {
        return false;
      }
    }
  }
  return false;
}


//Input: an array of integers, representing a list of MNOLS
//Output: true if the array of integers represents a set-canonical list of
    MNOLS.
//Method: We create every list of MNOLS is the same set-canonical as this list
    of MNOLS, and check this list is the least under the canonical form
    described in the chapter
//Note: other variables are tools for the calculations, either for the array
    size or as pre-defined memmory.
bool Check_Canonical(unsigned __int64** Z, unsigned __int64 n, unsigned
    __int64 k, unsigned __int64** B1, unsigned __int64** C1, unsigned
    __int64** D1, unsigned __int64** E1) {

  bool result = true;
  for (int alpha = 0; alpha < n; alpha++) {

    for (int i = 0; i < n; i++) {
      for (int j = 0; j < k; j++) {
        B1[j][i] = (Z[j][i] + alpha) % n;
      }
    }

    //apply m_x to B to get C
    for (int x = 1; x < n; x++) {

      //we require gcd(x,n)=1
      if (GCD(x, n) == 1) {
        for (int i = 0; i < n; i++) {
          for (int j = 0; j < k; j++) {
            E1[j][i] = (B1[j][i] * x) % n;
          }
        }

                //permuting the order of the list
        Permutation sigma(k);
        for (sigma.reset(); sigma.notFinished(); sigma.next()) {
          for (int i = 0; i < n; i++) {
            for (int j = 0; j < k; j++) {
              C1[sigma.get(j)][i] = E1[j][i];
            }
          }

          //Reorder so the list is in reduced order
          for (int i = 0; i < k; i++) {
```

```
          for (int j = 0; j < n; j++) {
            D1[i][C1[0][j]] = C1[i][j];
          }
        }

                    //fail if the resulting list of MNOLS is lower than the
                        original
        if (lower(D1, Z, n, k)) {
          result = false;
        }
      }

      sigma.del();

    }
  }
}
return result;
}




//Input: two arrays of integers, each representing a list of MNOLS
//Output: true if the two arrays are identicle, false otherwise
bool equal(unsigned __int64** Compare, unsigned __int64** Base, int n, int k) {

  for (int i = 0; i<k; i++) {
    for (int j = 0; j < n; j++) {
      if (Compare[i][j] != Base[i][j]) {
        return false;
      }
    }
  }
  //equal
  return true;
}


//Input: an array of integers, representing a list of MNOLS, which we denote
    as L
//Output (via the input pointer 'results'): true if the array of integers
    represents a set-canonical list of MNOLS.
//Method: We check which group actions are in Is_s(L),
    Is_l(L),Red_s(L),Red_l(L)
//Note: other variables are tools for the calculations, either for the array
    size or as pre-defined memmory.
void Count_Autotopisms(unsigned __int64** Z, unsigned __int64** B, unsigned
    __int64** C, unsigned __int64** D, unsigned __int64**E, unsigned
    __int64**Reorder, int n, int k, unsigned __int64* results) //bool&
    reset_to_fini, int***Store, int*count_MNOLS, vector<Holder>& MNOLS3,
{

  for (int i = 0; i < 5; i++){
    results[i] = 0;
  }
  //start searching through setw, with and without reduction
  unsigned __int64 S_count = 0;
  unsigned __int64 RS_count = 0;
  Permutation sigma(k);
  for (sigma.reset(); sigma.notFinished(); sigma.next()) {
```

```
    for (int i = 0; i<k; i++) {
      for (int j = 0; j < n; j++) {
        C[i][j] = Z[sigma.get(i)][j] % n;
      }
    }

    if (equal(C, Z, n, k)) {
      S_count++;
    }
    //place the MNOLS into reduced order
    for (int i = 0; i<k; i++) {
      for (int j = 0; j < n; j++) {
        D[i][C[0][j]] = C[i][j] % n;
      }
    }

    if (equal(D, Z, n, k)) {
      RS_count++;
    }
  }
  sigma.del();
  results[3] = RS_count; //the size of Red_s(L)
  results[4] = S_count;  //the size of Set(L)

  unsigned __int64 MTR_count = 0;
  unsigned __int64 MTRS_count = 0;

  //apply tau^alpha
  for (int alpha = 0; alpha<n; alpha++) {
    for (int i = 0; i<k; i++) {
      for (int j = 0; j < n; j++) {
        E[i][j] = (Z[i][j] + alpha) % n;
      }
    }

    //apply m_x to B
    for (int x = 1; (x<n); x++) { // && (reset_to_fini == false)

      //we require gcd(x,n)=1
      if (GCD(x, n) == 1) {
        for (int i = 0; i<k; i++) {
          for (int j = 0; j < n; j++) {
            B[i][j] = (E[i][j] * x) % n;
          }
        }

        //place the MNOLS into reduced order
        for (int i = 0; i<k; i++) {
          for (int j = 0; j < n; j++) {
            Reorder[i][B[0][j]] = B[i][j] % n;
          }
        }
        if (equal(Reorder, Z, n, k)) {
          MTR_count++;
        }

        Permutation sigma(k);
        for (sigma.reset(); sigma.notFinished(); sigma.next()) {

          for (int i = 0; i<k; i++) {
```

```
          for (int j = 0; j < n; j++) {
            C[i][j] = B[sigma.get(i)][j] % n;
          }
        }

        //place the MNOLS into reduced order
        for (int i = 0; i<k; i++) {
          for (int j = 0; j < n; j++) {
            D[i][C[0][j]] = C[i][j] % n;
          }
        }
        if (equal(D, Z, n, k)) {
          MTRS_count++;
        }
      }
      sigma.del();
    }
  }
}

if (MTRS_count == 0){
  cout << "we detected it here!\n";
  for (int i = 0; i<k; i++) {
    for (int j = 0; j < n; j++) {
      cout << Z[i][j] << ' ';
    }
    cout << '\n';
  }
}

results[0] = 1;
results[1] = MTRS_count; //the size of Is_s(L)
results[2] = MTR_count; //the size of Is_l(L)
return;
}



// we run this whole program for an individualy choice of the order of the
   Latin squares n
#define n 16

int main()
{

  omp_lock_t writelock3;
  omp_lock_t writelock4;
  omp_lock_t writelock5;

  omp_init_lock(&writelock3);
  omp_init_lock(&writelock4);
  omp_init_lock(&writelock5);


  int MNOLS_counter2 = 0;
  int* count = new int[n];
  for (int i = 0; i < n; i++) {
    count[i] = 0;
  }
```

```
  cout << n;

  vector<vector<__int8>> Canonical_Columns;
  vector<vector<__int8>> Potential_Third_Columns;

  vector<vector<vector<__int8>>> Canonical_Columns_partial(n);
  vector<vector<vector<__int8>>> Potential_Third_Columns_partial(n);

  unsigned __int64 COUNT_2way_Total = 0;
  unsigned __int64 COUNT_2way_Canonical = 0;


//For each mu, we create an array for the results of each type
// For each list of MNOLS in set-canonical form, L, of type t we perform:
    StoreResults(mu)Type(t)[Is_s(L)][Is_l(L)][Red_s(l)]++;
  unsigned __int64*** StoreResults2Type0;
  unsigned __int64*** StoreResults2Type1;
  unsigned __int64*** StoreResults3Type0;
  unsigned __int64*** StoreResults3Type1;
  unsigned __int64*** StoreResults4Type0;
  unsigned __int64*** StoreResults4Type1;
  unsigned __int64*** StoreResults5Type0;
  unsigned __int64*** StoreResults5Type1;

  StoreResults2Type0 = new unsigned __int64**[n*n];
  StoreResults2Type1 = new unsigned __int64**[n*n];
  StoreResults3Type0 = new unsigned __int64**[n*n];
  StoreResults3Type1 = new unsigned __int64**[n*n];
  StoreResults4Type0 = new unsigned __int64**[n*n];
  StoreResults4Type1 = new unsigned __int64**[n*n];
  StoreResults5Type0 = new unsigned __int64**[n*n];
  StoreResults5Type1 = new unsigned __int64**[n*n];
  for (int j = 0; j < n*n; j++){
    StoreResults2Type0[j] = new unsigned __int64*[6 + 1];
    StoreResults2Type1[j] = new unsigned __int64*[6 + 1];
    StoreResults3Type0[j] = new unsigned __int64*[6 + 1];
    StoreResults3Type1[j] = new unsigned __int64*[6 + 1];
    StoreResults4Type0[j] = new unsigned __int64*[6 + 1];
    StoreResults4Type1[j] = new unsigned __int64*[6 + 1];
    for (int k = 0; k < 6 + 1; k++){
      StoreResults2Type0[j][k] = new unsigned __int64[6 + 1];
      StoreResults2Type1[j][k] = new unsigned __int64[6 + 1];
      StoreResults3Type0[j][k] = new unsigned __int64[6 + 1];
      StoreResults3Type1[j][k] = new unsigned __int64[6 + 1];
      for (int m = 0; m < 6 + 1; m++){
        StoreResults2Type0[j][k][m] = 0;
        StoreResults2Type1[j][k][m] = 0;
        StoreResults3Type0[j][k][m] = 0;
        StoreResults3Type1[j][k][m] = 0;
      }
    }
  }
  for (int j = 0; j < n*n; j++){
    StoreResults4Type0[j] = new unsigned __int64*[24 + 1];
    StoreResults4Type1[j] = new unsigned __int64*[24 + 1];
    for (int k = 0; k < 24 + 1; k++){
      StoreResults4Type0[j][k] = new unsigned __int64[24 + 1];
      StoreResults4Type1[j][k] = new unsigned __int64[24 + 1];
      for (int m = 0; m < 24 + 1; m++){
        StoreResults4Type0[j][k][m] = 0;
```

```
        StoreResults4Type1[j][k][m] = 0;
      }
    }
  }
  for (int j = 0; j < n*n; j++){
    StoreResults5Type0[j] = new unsigned __int64*[120 + 1];
    StoreResults5Type1[j] = new unsigned __int64*[120 + 1];
    for (int k = 0; k < 120 + 1; k++){
      StoreResults5Type0[j][k] = new unsigned __int64[120 + 1];
      StoreResults5Type1[j][k] = new unsigned __int64[120 + 1];
      for (int m = 0; m < 120 + 1; m++){
        StoreResults5Type0[j][k][m] = 0;
        StoreResults5Type1[j][k][m] = 0;
      }
    }
  }


//count_2MNOLS stores how many
    [1]set-isotopy/[2]list-isotopy/[3]set-reduced/[4]list-reduced/[5]set
    classes and [6]lists in C^n_(mu)
  unsigned __int64 **count_2MNOLS;
  count_2MNOLS = new unsigned __int64*[n + 1];
  for (int i = 0; i < n + 1; i++) {
    count_2MNOLS[i] = new unsigned __int64[6]; //long int
    for (int j = 0; j < 6; j++){
      count_2MNOLS[i][j] = 0;
    }
  }



//The primary function of this section is to develop Canonical_Columns and
    Potential_Third_Columns
//  Canonical_Columns form the possible second columns of a list is MNOLS that
    are set-isotopic
//  Potential_Third_Columns the possible third/fourth/fifth columns of a list
    is MNOLS that are set-isotopic
  unsigned __int64 start1 = (std::clock());
  int max_k = 5;
  omp_set_num_threads(n);
#pragma omp parallel for
  for (int first_pos = 0; first_pos < n; first_pos++){

///////////set up local variable//////////
    unsigned __int64 temp[n];
    unsigned __int64 temp2[n];
    bool reset_to_fini = false;
    Permutation I(n);
    Permutation A(n);
    //Permutation H(n);
    unsigned __int64* temp_all = new unsigned __int64[n];
    //for (Permutation A(n); A.notFinished() == true; A.next()){

    unsigned __int64* results = new unsigned __int64[5];

    unsigned __int64**Z = new unsigned __int64*[max_k];
    unsigned __int64** BBB = new unsigned __int64*[max_k];
    unsigned __int64** CCC = new unsigned __int64*[max_k];
    unsigned __int64** DDD = new unsigned __int64*[max_k];
    unsigned __int64** EEE = new unsigned __int64*[max_k];
```

```
    unsigned __int64** FFF = new unsigned __int64*[max_k];

    for (int i = 0; i < max_k; i++) {
      Z[i] = new unsigned __int64[n];
      BBB[i] = new unsigned __int64[n];
      CCC[i] = new unsigned __int64[n];
      DDD[i] = new unsigned __int64[n];
      EEE[i] = new unsigned __int64[n];
      FFF[i] = new unsigned __int64[n];
      for (int j = 0; j < n; j++) {
        Z[i][j] = 0;
        BBB[i][j] = 0;
        CCC[i][j] = 0;
        DDD[i][j] = 0;
        EEE[i][j] = 0;
        FFF[i][j] = 0;
      }
    }
////////////finished defining local variable//////////

    //Increase the permutation A until the first entry is first pos
    while (A.get(0) != first_pos){
      A.next(0);
    }

    //repeat the following until all permutations starting with 'first_pos'
        are exhausted
    while (A.notFinished() && A.get(0) == first_pos) {
    reset_to_fini = false;

    //Check this permutation 'A' is nearly orthogonal to the identity
        permutation
    if (reset_to_fini == false) {
      if (checkMNOLS(I, A, temp_all) == false) {
        reset_to_fini = true;
      }
    }


//set up some variable for later
    if (reset_to_fini == false) {
      //A.Print();

      ///////////////////////////////////////check if canonical 2MNOLS
      for (int i = 0; i < max_k; i++) {
        for (int j = 0; j < n; j++) {
          BBB[i][j] = 0;
          CCC[i][j] = 0;
          DDD[i][j] = 0;
          EEE[i][j] = 0;
        }
      }
      for (int i = 0; i < n; i++) {
        Z[0][i] = i;
        Z[1][i] = A.get(i);
      }

      //If we finished this without setting reset_to_fini =true, then (I,A)
          represents a possible list of MNOLS
    }
```

```
    //if the permutations (I,A) are a list of MNOLS
    if (reset_to_fini == false) {
      if (COUNT_2way_Total % 1000 == 0){
        cout << COUNT_2way_Total << " ";
      }
      if (COUNT_2way_Total % 10000 == 0){
        A.Print();
      }
      //if A is a potential second column in a list of MNOLS that are
          set-canonical (implies first entry is 1 & canonical)
      if (A.get(0) == 1){
        if (Check_Canonical(Z, n, 2, BBB, CCC, DDD, EEE)){

          //count how many lists of 2 cyclic MNOLS under each equivalence that
              correspond to the set-canonical list of MNOLS (I,A)
          Count_Autotopisms(Z, BBB, CCC, DDD, EEE, FFF, n, 2, results);
          if (results[1] != 0){
#pragma omp atomic
            count_2MNOLS[n][0] += results[0];
#pragma omp atomic
            count_2MNOLS[n][1] += totient(n) * n * results[3] / results[1];
#pragma omp atomic
            count_2MNOLS[n][2] += totient(n) * n * factorial(2) * 1 /
                results[1];
#pragma omp atomic
            count_2MNOLS[n][3] += factorial(2) * results[2] / results[1];
#pragma omp atomic
            count_2MNOLS[n][4] += totient(n) * n * results[4] / results[1];
                //divided by n!
#pragma omp atomic
            count_2MNOLS[n][5] += totient(n) * n * factorial(2) / results[1];
                //divided by n!

          }

          //Store based on types
          if (Z[1][1] == 0){
#pragma omp atomic
            StoreResults2Type0[results[1]][results[2]][results[3]]++;
          }
          else{
#pragma omp atomic
            StoreResults2Type1[results[1]][results[2]][results[3]]++;
          }


          //Save this column in a vector of all possible second columns
          vector<__int8> temp;
          for (int i = 0; i < n; i++){
            temp.push_back(A.get(i));
          }
          Canonical_Columns_partial[first_pos].push_back(temp);
#pragma omp atomic
          COUNT_2way_Canonical++;
        }
#pragma omp atomic
        COUNT_2way_Total++;
      }
      else{
        //otherwise, this column could be the third/fourth/etc column
```

```
        //Save this column in a vector of all possible third/etc columns
        vector<__int8> temp;
        for (int i = 0; i < n; i++){
          temp.push_back(A.get(i));
        }
        Potential_Third_Columns_partial[first_pos].push_back(temp);
#pragma omp atomic
        COUNT_2way_Total++;
      }
      A.next();
    }
  }

}

//As we have done this over several threads, put spread out lists into a
    common list
for (int first_pos = 0; first_pos < n; first_pos++){
  for (int i = 0; i < Canonical_Columns_partial[first_pos].size(); i++){
    Canonical_Columns.push_back(Canonical_Columns_partial[first_pos][i]);
  }
  Canonical_Columns_partial[first_pos].clear();
  for (int i = 0; i < Potential_Third_Columns_partial[first_pos].size(); i++){
    Potential_Third_Columns.push_back(Potential_Third_Columns_partial[first_pos][i]);
  }
  Potential_Third_Columns_partial[first_pos].clear();
}
Canonical_Columns_partial.clear();
Potential_Third_Columns_partial.clear();



  //the list is produced
  cout << "the number of second columns is " << COUNT_2way_Total << " and " <<
      COUNT_2way_Canonical << '\n';
  cout << "the list Potential_Third_Columns has size " <<
      Potential_Third_Columns.size() << " and Canonical_Columns has size " <<
      Canonical_Columns.size() << ", which is produced in time: " <<
      (std::clock() - start1) / (double)CLOCKS_PER_SEC << '\n';




//create variables to store results
  unsigned __int64 COUNT_3way_Canonical = 0;
  unsigned __int64 COUNT_4way_Canonical = 0;
  unsigned __int64 COUNT_5way_Canonical = 0;

  unsigned __int64 **count_3MNOLS;
  count_3MNOLS = new unsigned __int64*[n + 1];
  for (int i = 0; i < n + 1; i++) {
    count_3MNOLS[i] = new unsigned __int64[6];
    for (int j = 0; j < 6; j++){
      count_3MNOLS[i][j] = 0;
    }
  }
  unsigned __int64 **count_4MNOLS;
  count_4MNOLS = new unsigned __int64*[n + 1];
  for (int i = 0; i < n + 1; i++) {
```

```
    count_4MNOLS[i] = new unsigned __int64[6];
    for (int j = 0; j < 6; j++){
      count_4MNOLS[i][j] = 0;
    }
  }
  unsigned __int64 **count_5MNOLS;
  count_5MNOLS = new unsigned __int64*[n + 1];
  for (int i = 0; i < n + 1; i++) {
    count_5MNOLS[i] = new unsigned __int64[6];
    for (int j = 0; j < 6; j++){
      count_5MNOLS[i][j] = 0;
    }
  }
//create an output list for the lists of MNOLS
  string file_name_3 = "3MNOLS-n";
  file_name_3 += to_string(_Longlong(n));
  file_name_3 += ".txt";

  string file_name_4 = "4MNOLS-n";
  file_name_4 += to_string(_Longlong(n));
  file_name_4 += ".txt";


  string file_name_5 = "5MNOLS-n";
  file_name_5 += to_string(_Longlong(n));
  file_name_5 += ".txt";

  std::ofstream file_to_be_appended;
  file_to_be_appended.open(file_name_3, ios::out);
  file_to_be_appended.close();
  file_to_be_appended.open(file_name_4, ios::out);
  file_to_be_appended.close();
  file_to_be_appended.open(file_name_5, ios::out);
  file_to_be_appended.close();



  int count_doneThreads = 0;
  struct pair_vec
  {
    vector<__int8> first, second;
  };
  vector<pair_vec> List_all;
  bool reset_to_fini_1 = false;


//for each possible second column...
  omp_set_num_threads(n);
#pragma omp parallel for // ordered schedule(dynamic, 1)
  for (int first = 0; first < Canonical_Columns.size(); first++) {

    cout << count_doneThreads++ << ' ';

    //Every 1000 threads, we output some details as to the current state of
       the program
    if (count_doneThreads % 1000 == 0){

      cout << "time = " << (std::clock() - start1) / (double)CLOCKS_PER_SEC <<
         '\n';


      cout << " \n mu=2\n";
```

```
  cout << "total set-iso classes: " << count_2MNOLS[n][0] << "\n";
  cout << "total set-red classes: " << count_2MNOLS[n][1] << "\n";
  cout << "total list-red classes: " << count_2MNOLS[n][2] << "\n";
  cout << "total list-iso classes: " << count_2MNOLS[n][3] << "\n";
  cout << "total set classes (divided by n!): " << count_2MNOLS[n][4] <<
      "\n";
  cout << "total lists (divided by n!): " << count_2MNOLS[n][5] << "\n";
  cout << '\n';

  cout << " \n mu=3\n";
  cout << "total set-iso classes: " << count_3MNOLS[n][0] << "\n";
  cout << "total set-red classes: " << count_3MNOLS[n][1] << "\n";
  cout << "total list-red classes: " << count_3MNOLS[n][2] << "\n";
  cout << "total list-iso classes: " << count_3MNOLS[n][3] << "\n";
  cout << "total set classes (divided by n!): " << count_3MNOLS[n][4] <<
      "\n";
  cout << "total lists (divided by n!): " << count_3MNOLS[n][5] << "\n";
  cout << '\n';


  cout << " mu=4\n";
  cout << "total set-iso classes: " << count_4MNOLS[n][0] << "\n";
  cout << "total set-red classes: " << count_4MNOLS[n][1] << "\n";
  cout << "total list-red classes: " << count_4MNOLS[n][2] << "\n";
  cout << "total list-iso classes: " << count_4MNOLS[n][3] << "\n";
  cout << "total set classes (divided by n!): " << count_4MNOLS[n][4] <<
      "\n";
  cout << "total lists (divided by n!): " << count_4MNOLS[n][5] << "\n";
  cout << '\n';


  cout << " mu=5\n";
  cout << "total set-iso classes: " << count_5MNOLS[n][0] << "\n";
  cout << "total set-red classes: " << count_5MNOLS[n][1] << "\n";
  cout << "total list-red classes: " << count_5MNOLS[n][2] << "\n";
  cout << "total list-iso classes: " << count_5MNOLS[n][3] << "\n";
  cout << "total set classes (divided by n!): " << count_5MNOLS[n][4] <<
      "\n";
  cout << "total lists (divided by n!): " << count_5MNOLS[n][5] << "\n";
  cout << '\n';
}

//define local variables, for use in functions without having to redefine
    variables
unsigned __int64* results = new unsigned __int64[5];
reset_to_fini_1 = false;
unsigned __int64**Z = new unsigned __int64*[max_k];
unsigned __int64** BBB = new unsigned __int64*[max_k];
unsigned __int64** CCC = new unsigned __int64*[max_k];
unsigned __int64** DDD = new unsigned __int64*[max_k];
unsigned __int64** EEE = new unsigned __int64*[max_k];
unsigned __int64** FFF = new unsigned __int64*[max_k];
for (int i = 0; i < max_k; i++) {
  Z[i] = new unsigned __int64[n];
  BBB[i] = new unsigned __int64[n];
  CCC[i] = new unsigned __int64[n];
  DDD[i] = new unsigned __int64[n];
  EEE[i] = new unsigned __int64[n];
  FFF[i] = new unsigned __int64[n];
  for (int j = 0; j < n; j++) {
    Z[i][j] = 0;
```

```
    BBB[i][j] = 0;
    CCC[i][j] = 0;
    DDD[i][j] = 0;
    EEE[i][j] = 0;
    FFF[i][j] = 0;
  }
}

bool reset_to_fini_2 = false;
vector<vector<__int8>> List;
vector<vector<__int8>> List_3rdColumn;
vector<vector<__int8>>::const_iterator i2;
vector<vector<__int8>>::const_iterator i3;

//Fill the possible columns into the permutations P1, etc.
Permutation P1(n), P2(n), P3(n);
for (int i = 0; i < n; i++) {
  P1.set(i, Canonical_Columns[first][i]);
}

unsigned __int64* temp_all = new unsigned __int64[n];



//reset local variables
for (int i = 0; i < max_k; i++) {
  for (int j = 0; j < n; j++) {
    BBB[i][j] = 0;
    CCC[i][j] = 0;
    DDD[i][j] = 0;
    EEE[i][j] = 0;
  }
}
for (int i = 0; i < n; i++) {
  Z[0][i] = i;
  Z[1][i] = Canonical_Columns[first][i];
}

//we now move onto the third column
List.empty();
List_3rdColumn.empty();
if (reset_to_fini_1 == false){
  for (unsigned __int64 second = 0; second <
      Potential_Third_Columns.size(); second++) {

    //The permutation object P2 holds the third permutation
    for (int i = 0; i < n; i++) {
      P2.set(i, Potential_Third_Columns[second][i]);
    }

    //check that P1 and P2 (thinking of them as columns) are nearly
        orthogonal
    if (checkMNOLS(P1, P2, temp_all) == 1) {
      reset_to_fini_2 = false;

      //load P2 into Z for Check_Canonical and Count_Autotopisms later
      for (int i = 0; i < n; i++) {
        Z[2][i] = Potential_Third_Columns[second][i];
      }
```

```
        //develops List, which is all posiible third/fourth/fifth columns
        if (reset_to_fini_2 == false) {
          vector<__int8> temp;
          for (int i = 0; i < n; i++) {
            temp.push_back(Z[2][i]);
            //temp[i] = Z[2][i];
          }
          List.push_back(temp);
        }

        //if this one so happen to be canonical, then we have a list of 3
            cyclic MNOLS that are set-canonical
        if (reset_to_fini_2 == false) {
          if (Check_Canonical(Z, n, 3, BBB, CCC, DDD, EEE) == true) {
          vector<__int8> temp;
          for (int i = 0; i < n; i++) {
            temp.push_back(Z[2][i]);
            //temp[i] = Z[2][i];
          }
          List_3rdColumn.push_back(temp);

          //store results
#pragma omp atomic
            COUNT_3way_Canonical++;

            Count_Autotopisms(Z, BBB, CCC, DDD, EEE, FFF, n, 3, results);
            if (results[1] != 0){

#pragma omp atomic
              count_3MNOLS[n][0] += results[0];
#pragma omp atomic
              count_3MNOLS[n][1] += totient(n) * n * results[3] / results[1];
#pragma omp atomic
              count_3MNOLS[n][2] += totient(n) * n * factorial(3) * 1 /
                  results[1];
#pragma omp atomic
              count_3MNOLS[n][3] += factorial(3) * results[2] / results[1];
#pragma omp atomic
              count_3MNOLS[n][4] += totient(n) * n * results[4] / results[1];
                  //divided by n!
#pragma omp atomic
              count_3MNOLS[n][5] += totient(n) * n * factorial(3) /
                  results[1]; //divided by n! // * results[4] was here

            }

            //store type information
            if (Z[1][1] == 0){
#pragma omp atomic
              StoreResults3Type0[results[1]][results[2]][results[3]]++;
            }
            else{
#pragma omp atomic
              StoreResults3Type1[results[1]][results[2]][results[3]]++;
            }
          }
        }
      }
    }

//We now come to the part of the program where we create a graph and look for
    a clique.
```

```
//As we mentioned previously, we have unrolled the clique problem

//At the present time, the first and second columns are fixed.
      Permutation temp1(n), temp2(n), temp3(n);

      //we develop temp1 (column three) and temp2 (column four)
      for (unsigned __int64 i = 0; i < List_3rdColumn.size(); i++){
        for (int k = 0; k < n; k++){
          temp1.set(k, List_3rdColumn[i][k]);
        }
        for (unsigned __int64 j = 0; j < List.size(); j++){
          for (int k = 0; k < n; k++){
            temp2.set(k, List[j][k]);
          }
          //if the third and fourth columns are nearly orthogonal
          if (checkMNOLS(temp1, temp2, temp_all) == true){
            //we currently have a list of 4 cyclic MNOLS

            for (int l = 0; l < n; l++) {
              Z[2][l] = List_3rdColumn[i][l];
              Z[3][l] = List[j][l];
            }
            if (Check_Canonical(Z, n, 4, BBB, CCC, DDD, EEE) == true) {

              //If we get here, we have found a list of 4 cyclic MNOLS that are
                  set-isotopic, so store the results
              #pragma omp atomic
              COUNT_4way_Canonical++;

              Count_Autotopisms(Z, BBB, CCC, DDD, EEE, FFF, n, 4, results);
              if (results[1] != 0){
#pragma omp atomic
                count_4MNOLS[n][0] += results[0];
#pragma omp atomic
                count_4MNOLS[n][1] += totient(n) * n * results[3] / results[1];
#pragma omp atomic
                count_4MNOLS[n][2] += totient(n) * n * factorial(4) * 1 /
                    results[1];
#pragma omp atomic
                count_4MNOLS[n][3] += factorial(4) * results[2] / results[1];
#pragma omp atomic
                count_4MNOLS[n][4] += totient(n) * n * results[4] / results[1];
                    //divided by n!
#pragma omp atomic
                count_4MNOLS[n][5] += totient(n) * n * factorial(4) /
                    results[1]; //divided by n! // * results[4] was here

              }

              if (Z[1][1] == 0){
#pragma omp atomic
                StoreResults4Type0[results[1]][results[2]][results[3]]++;
              }
              else{
#pragma omp atomic
                StoreResults4Type1[results[1]][results[2]][results[3]]++;
              }

              //print to file
              string found_MNOLS = "";
              for (int i = 0; i < 4; i++) {
                for (int j = 0; j < n; j++) {
                  found_MNOLS.append(to_string(Z[i][j]));
```

```
          found_MNOLS.append(" ");
        }
        found_MNOLS.append("\n");
      }
      found_MNOLS.append("\n");

      omp_set_lock(&writelock4);
      file_to_be_appended.open(file_name_4, ios::out | ios::app);
      file_to_be_appended << found_MNOLS;
      file_to_be_appended.close();
      omp_unset_lock(&writelock4);
      found_MNOLS.clear();

      //At this point, we could store this third and fourth column as
          an edge in a graph.
      //Instead, we search for a fifth column to form a list of 5
          cyclic MNOLS that are set-isotopic
      for (unsigned __int64 k = 0; k < List.size(); k++){
        for (int l = 0; l < n; l++){
          temp3.set(l, List[k][l]);
        }
        if (checkMNOLS(temp1, temp3, temp_all) == true){
          if (checkMNOLS(temp2, temp3, temp_all) == true){
            for (int l = 0; l < n; l++) {
              Z[4][l] = List[k][l];
            }
            if (Check_Canonical(Z, n, 5, BBB, CCC, DDD, EEE) == true) {

            //we have found a list of 5 cyclic MNOLS that are
                set-isotopic, so store the results
              if (COUNT_5way_Canonical==0){
                cout << "we have a five way!\n";
                for (int a = 0; a < 5; a++){
                  for (int b = 0; b < n; b++){
                    cout << Z[a][b] << ' ';
                  }
                  cout << '\n';
                }
                cout << '\n';
              }
              Count_Autotopisms(Z, BBB, CCC, DDD, EEE, FFF, n, 5,
                  results);
#pragma omp atomic
              count_5MNOLS[n][0] += results[0];
#pragma omp atomic
              count_5MNOLS[n][1] += totient(n) * n * results[3] /
                  results[1];
#pragma omp atomic
              count_5MNOLS[n][2] += totient(n) * n * factorial(5) * 1 /
                  results[1];
#pragma omp atomic
              count_5MNOLS[n][3] += factorial(5) * results[2] /
                  results[1];
#pragma omp atomic
              count_5MNOLS[n][4] += totient(n) * n * results[4] /
                  results[1]; //divided by n!
#pragma omp atomic
              count_5MNOLS[n][5] += totient(n) * n * factorial(5) /
                  results[1]; //divided by n! // * results[4] was here

#pragma omp atomic
```

```
                        COUNT_5way_Canonical++;
                        if (Z[1][1] == 0){
#pragma omp atomic
                          StoreResults5Type0[results[1]][results[2]][results[3]]++;
                        }
                        else{
#pragma omp atomic
                          StoreResults5Type1[results[1]][results[2]][results[3]]++;
                        }

                        //print to file
                        string found_MNOLS = "";
                        for (int i = 0; i < 5; i++) {
                          for (int j = 0; j < n; j++) {
                            found_MNOLS.append(to_string(Z[i][j]));
                            found_MNOLS.append(" ");
                          }
                          found_MNOLS.append("\n");
                        }
                        found_MNOLS.append("\n");

                        omp_set_lock(&writelock5);
                        file_to_be_appended.open(file_name_5, ios::out | ios::app);
                        file_to_be_appended << found_MNOLS;
                        file_to_be_appended.close();
                        omp_unset_lock(&writelock5);
                        found_MNOLS.clear();
                      }
                    }
                  }
                }

              }
            }
          }
        }
      }

//clear stored data
    List.clear();
    List.resize(0);

    for (int i = 0; i < max_k; i++) {
      delete[] Z[i];
      delete[] BBB[i];
      delete[] CCC[i];
      delete[] DDD[i];
      delete[] EEE[i];
    }
    delete[] Z;
    delete[] BBB;
    delete[] CCC;
    delete[] DDD;
    delete[] EEE;
    List.clear();
    delete[] temp_all;
```

```
  }



//Print results
  bool print_m;
  bool print_j;

  cout << "\n";

  cout << "mu=2\n";

  for (int m = 0; m < 6 + 1; m++){
    print_m = false;
    for (int j = 0; j < n*n; j++){
      for (int k = 0; k < 6 + 1; k++){
        if (StoreResults2Type0[j][k][m] > 0 || StoreResults2Type1[j][k][m] >
            0){
          print_m = true;
          cout << "(is_s {j},is_l {k},red_s {m},t1,t2,all) = (" << j << ',' <<
              k << ',' << m << ',' << StoreResults2Type0[j][k][m] << ',' <<
              StoreResults2Type1[j][k][m] << ',' << StoreResults2Type0[j][k][m]
              + StoreResults2Type1[j][k][m] << ")\n";
        }
      }
    }
  }
  cout << '\n';

  cout << "mu=3\n";

    for (int m = 0; m < 6 + 1; m++){
      print_m = false;
      for (int j = 0; j < n*n; j++){
        for (int k = 0; k < 6 + 1; k++){
          if (StoreResults3Type0[j][k][m] > 0 || StoreResults3Type1[j][k][m] >
              0){
            print_m = true;
            cout << "(is_s {j},is_l {k},red_s {m},t1,t2,all) = (" << j << ','
                << k << ',' << m<< ',' << StoreResults3Type0[j][k][m] << ',' <<
                StoreResults3Type1[j][k][m] << ',' <<
                StoreResults3Type0[j][k][m] + StoreResults3Type1[j][k][m] <<
                ")\n";
          }
        }
      }
    }
    cout << '\n';


  cout << "mu=4\n";
    for (int m = 0; m < 24 + 1; m++){
      print_m = false;
      for (int j = 0; j < n*n; j++){
        for (int k = 0; k < 24 + 1; k++){
          if (StoreResults4Type0[j][k][m] > 0 || StoreResults4Type1[j][k][m] >
              0){
            print_m = true;
```

```
          cout << "(is_s {j},is_l {k},red_s {m},t1,t2,all) = (" << j << ','
             << k << ',' << m << ',' << StoreResults4Type0[j][k][m] << ','
             << StoreResults4Type1[j][k][m] << ',' <<
             StoreResults4Type0[j][k][m] + StoreResults4Type1[j][k][m] <<
             ")\n";
        }
      }
    }
  }
  cout << '\n';


int max_j;
cout << "mu=5\n";

  for (int m = 0; m < 120 + 1; m++){
    max_j = 0;
    print_m = false;
    for (int j = 0; j < n*n; j++){
      for (int k = 0; k < 120 + 1; k++){
        if (StoreResults5Type0[j][k][m] > 0 || StoreResults5Type1[j][k][m] >
            0){
          print_m = true;
          cout << "(is_s {j},is_l {k},red_s {m},t1,t2,all) = (" << j << ','
             << k << ',' << m << ',' << StoreResults5Type0[j][k][m] << ','
             << StoreResults5Type1[j][k][m] << ',' <<
             StoreResults5Type0[j][k][m] + StoreResults5Type1[j][k][m] <<
             ")\n";
        }
      }
    }
  }
  cout << '\n';




  cout << " \n mu=2\n";
  cout << "total set-iso classes: " << count_2MNOLS[n][0] << "\n";
  cout << "total set-red classes: " << count_2MNOLS[n][1] << "\n";
  cout << "total list-red classes: " << count_2MNOLS[n][2] << "\n";
  cout << "total list-iso classes: " << count_2MNOLS[n][3] << "\n";
  cout << "total set classes (divided by n!): " << count_2MNOLS[n][4] <<
      "\n";
  cout << "total lists (divided by n!): " << count_2MNOLS[n][5] << "\n";
  cout << '\n';

cout << " \n mu=3\n";
  cout << "total set-iso classes: " << count_3MNOLS[n][0] << "\n";
  cout << "total set-red classes: " << count_3MNOLS[n][1] << "\n";
  cout << "total list-red classes: " << count_3MNOLS[n][2] << "\n";
  cout << "total list-iso classes: " << count_3MNOLS[n][3] << "\n";
  cout << "total set classes (divided by n!): " << count_3MNOLS[n][4] <<
      "\n";
  cout << "total lists (divided by n!): " << count_3MNOLS[n][5] << "\n";
cout << '\n';
```

```
  cout << " mu=4\n";
    cout << "total set-iso classes: " << count_4MNOLS[n][0] << "\n";
    cout << "total set-red classes: " << count_4MNOLS[n][1] << "\n";
    cout << "total list-red classes: " << count_4MNOLS[n][2] << "\n";
    cout << "total list-iso classes: " << count_4MNOLS[n][3] << "\n";
    cout << "total set classes (divided by n!): " << count_4MNOLS[n][4] <<
        "\n";
    cout << "total lists (divided by n!): " << count_4MNOLS[n][5] << "\n";
  cout << '\n';

  cout << " mu=5\n";
    cout << "total set-iso classes: " << count_5MNOLS[n][0] << "\n";
    cout << "total set-red classes: " << count_5MNOLS[n][1] << "\n";
    cout << "total list-red classes: " << count_5MNOLS[n][2] << "\n";
    cout << "total list-iso classes: " << count_5MNOLS[n][3] << "\n";
    cout << "total set classes (divided by n!): " << count_5MNOLS[n][4] <<
        "\n";
    cout << "total lists (divided by n!): " << count_5MNOLS[n][5] << "\n";
  cout << '\n';



  cout << "COUNT_2way_Canonical=" << COUNT_2way_Canonical<<'\n';
  cout << "COUNT_2way_Total=" << COUNT_2way_Total << '\n';
  cout << "COUNT_3way_Canonical" << COUNT_3way_Canonical << '\n';
  cout << "COUNT_4way_Canonical" << COUNT_4way_Canonical << '\n';
  cout << "COUNT_5way_Canonical" << COUNT_5way_Canonical << '\n';

  cout << "List_all size =" << List_all.size() << " ";
  cout << "MNOLS_counter2=" << MNOLS_counter2 << '\n';

  cout << "time = " << (std::clock() - start1) / (double)CLOCKS_PER_SEC <<
      '\n';

  return 0;
}
```