Near-Orthogonal Latin Squares From Neofields

Wei Hu; Supervisor: Lucia Moura, PhD

School of Electrical Engineering and Computer Science, University of Ottawa

Introduction

Sets of Mutually Orthogonal Latin Squares (MOLS) are important combinatorial designs with applications ranging from cryptography to software engineering. They can be used to form orthogonal arrays and covering arrays which allow the effective pairwise testing" or "t-wise testing" of systems.

factor values:	exhaustive testing	pairwise testing covering arrays
4 binary factors	16 tests	5 tests
12 factors 10 values each	1,000,000,000,000 tests	118 tests

Factors:	Operating System	Web browser	File format	Printer	binary outcome:
test1	1	1	1	1	PASS/FAIL
test2	1	0	0	0	PASS/FAIL
test3	0	1	0	0	PASS/FAIL
test4	0	0	1	0	PASS/FAIL
test5	0	0	0	1	PASS/FAIL

When the order is a prime power, it is known that the maximum number of MOLS that exists is one less than the order. This maximum set can be created using algebraic finite fields. However, relatively little is known about MOLS of other orders. Even for a “small” case like 10, it is an open question whether there exist 3 MOLS.

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Cayley Table of $(\mathbb{Z}_5, +)$

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

Latin Square of $(\mathbb{Z}_5, +)$

In 2004, Keedwell and Mullen [1], introduced the concept of a neofield which produces Latin Squares of even orders that are pairwise near-orthogonal (some pairs are missed, while others are covered more than once). In 2016, Dr. Daniel Droz, describes, in his thesis, new neofield constructions improving on the original results [2].

$\oplus_2$	∞	0	1	2	3	4	5	6	7	8
∞	∞	0	1	2	3	4	5	6	7	8
0	0	∞	2	4	6	8	1	3	5	7
1	1	8	∞	3	5	7	0	2	4	6
2	2	7	0	∞	4	6	8	1	3	5
3	3	6	8	1	∞	5	7	0	2	4
4	4	5	7	0	2	∞	6	8	1	3
5	5	4	6	8	1	3	∞	7	0	2
6	6	3	5	7	0	2	4	∞	8	1
7	7	2	4	6	8	1	3	5	∞	0
8	8	1	3	5	7	0	2	4	6	∞

Considering the novelty of the results, the idea is to explore the structure of the deficiencies in these sets of Near-Orthogonal Latin Squares, and to discover applications that make use of that structure.

Methodology

The project focused on *uniform cyclic neofields* which Dr. Daniel Droz devised in his thesis [2]. These neofields are defined as followed:

Definition 3.2 Let q be even, and let u be chosen so that $2 \leq u \leq q-2$ and $(u, q-1) = (u-1, q-1) = 1$. A *uniform cyclic neofield* of order q and character u is the neofield with the following properties:

- There are q elements, incuding additive and multiplicative identities 0 and 1.
- The multiplicative group of non-zero elements is cyclic of order $q-1$; we will call the generator θ so that the the elements of the neofield may be listed as $\{0, 1, \theta, \theta^2, \ldots, \theta^{q-2}\}$.
- The characteristic is 2; that is $a \oplus a = 0$ for any element a.
- We have, for all $1 \leq k \leq q-2$, $1 \oplus \theta^k = \theta^{uk}$.
- The fact that multiplication distributes over addition uniquely defines all other additions.

This neofield will be denoted $N_q^{(u)}$.

Algorithms were devised to simulate these mathematical relationships and allow neofields to be computationally generated.
The R Programming language was chosen for its extensive collection of packages and built-in functions for manipulating matrices.

```
mullen.neofield <- function(q, u, a) {
  m <- integer(q * q)
  m <- matrix(m , nrow = q, ncol = q)
  m[1, 1] <- Inf
  for (y in 0:(q - 2)) {
    m[1, 2 + y] <- y
  }
  for (x in 0:(q - 2)) {
    m[x + 2, 1] <- (x + a) %% (q - 1)
    for (y in 0:(q - 2)) {
      if ((x + a) %% (q - 1) == y) {
        m[x + 2, y + 2] <- Inf
      } else {
        # the coefficient "q * q" is to ensure %% is performed on a positive number
        m[x + 2, y + 2] <-
          (((a + x) + u * (y - (a + x))) + q * q * (q - 1)) %% (q - 1)
      }
    }
  }
```

A collections of R functions were devised to analyze, among other things, the deficiencies in coverage, the frequency and locations of duplicated coverage, the uniqueness of the generated Latin Squares.

Using these functions, the coverage claims made in the Droz thesis were verified, and the correctness of the algorithm above was ascertained.

Then, by observing the coverage, and noticing the presence of a cyclic pattern, the concept of a “starter” was devised.

> coverage(mullen.neofield(14, 5, 2), mullen.neofield(14, 4, 3))

```
    Inf 0 1 2 3 4 5 6 7 8 9 10 11 12
Inf   1 1 1 1 1 1 1 1 1 1  1  1  1
0     1 2 2 1 1 1 1 1 1 1 0  0  1  1
1     1 1 2 2 1 1 1 1 1 1 1  0  0  1
2     1 1 1 2 2 1 1 1 1 1 1  1  0  0
3     1 0 1 1 2 2 1 1 1 1 1  1  1  0
4     1 0 0 1 1 2 2 1 1 1 1  1  1  1
5     1 1 0 0 1 1 2 2 1 1 1  1  1  1
6     1 1 1 0 0 1 1 2 2 1 1  1  1  1
7     1 1 1 1 0 0 1 1 2 2 1  1  1  1
8     1 1 1 1 1 0 0 1 1 2 2  1  1  1
9     1 1 1 1 1 1 0 0 1 1 2  2  1  1
10    1 1 1 1 1 1 1 0 0 1 1  2  2  1
11    1 1 1 1 1 1 1 1 0 0 1  1  2  2
12    1 2 1 1 1 1 1 1 1 0 0  1  1  2
```

The starter is found on the second row (in this case “2211111110011”); each row that follows consists of the ”starter” cyclically shifted right. This property is then taken advantage of to make covering arrays.

Results

The algorithm for generating *uniform cyclic neofields* and the collection of R functions that were written can be found here: https://github.com/TheWeiHu/orthogonal-latin-squares

An algorithm was written to look for ways to group together neofields with “compatible starters” that would, in the process of making a covering array, allow a single added row to repair the deficiencies in multiple pairs of neofields.

	Generator A	Generator B	deficiency	starter	frequency	indices
1	14-03-00	14-02-00	26	c(2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	list(c(14, 1), c(2, 13), c(1, 156))	Inf
2	14-03-01	14-02-00	26	c(2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0)	list(c(1, 144), c(2, 26))	12:13
3	14-03-02	14-02-00	26	c(2, 1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 0)	list(c(1, 144), c(2, 26))	c(10, 12)
4	14-03-03	14-02-00	26	c(2, 1, 1, 2, 1, 1, 1, 0, 1, 1, 0, 1)	list(c(1, 144), c(2, 26))	c(8, 11)
5	14-03-04	14-02-00	26	c(2, 1, 1, 1, 2, 0, 1, 1, 1, 0, 1, 1)	list(c(1, 144), c(2, 26))	c(6, 10)
6	14-03-05	14-02-00	26	c(2, 1, 1, 0, 1, 2, 1, 1, 0, 1, 1, 1)	list(c(1, 144), c(2, 26))	c(4, 9)
7	14-03-06	14-02-00	26	c(2, 0, 1, 1, 1, 1, 2, 0, 1, 1, 1, 1)	list(c(1, 144), c(2, 26))	c(2, 8)
8	14-03-07	14-02-00	26	c(2, 1, 1, 1, 1, 1, 0, 2, 1, 1, 1, 0)	list(c(1, 144), c(2, 26))	c(7, 13)
9	14-03-08	14-02-00	26	c(2, 1, 1, 1, 1, 0, 1, 1, 2, 1, 0, 1)	list(c(1, 144), c(2, 26))	c(6, 11)
10	14-03-09	14-02-00	26	c(2, 1, 1, 1, 0, 1, 1, 1, 0, 2, 1, 1)	list(c(1, 144), c(2, 26))	c(5, 9)
11	14-03-10	14-02-00	26	c(2, 1, 1, 0, 1, 1, 0, 1, 1, 1, 2, 1)	list(c(1, 144), c(2, 26))	c(4, 7)
12	14-03-11	14-02-00	26	c(2, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 2)	list(c(1, 144), c(2, 26))	c(3, 5)
13	14-03-12	14-02-00	26	c(2, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2)	list(c(1, 144), c(2, 26))	2:3

Showing 1 to 14 of 18,590 entries

A tool to generate was created to generate all *uniform cyclic neofields* of a certain size as well as a table of their properties

Future Work

An interesting project would be to see if there is a way to improve on the current best of 228 for covering array CAN (2,11,14). The exist many sets of eleven neofields of size fourteen that have pairwise deficiencies of 26. If the diagonal which is covered twice can be “removed,” and sufficiently “compatible starters” can be found, there might be a way.

Also, it would be worthwhile to organize the the neofield algorithms and the collection of R functions that were written into an R Package

Acknowledgement

Professor Lucia Moura’s patience, guidance and passion has given me the best research experience I could have hope for. It has been a pleasure going into her office every week to explore new ideas, to gain from her insights, and to be taught by her.

I would also like to express my gratitude towards to the University of Ottawa UROP program for having granted me this opportunity. Furthermore, special thanks to Dr. Gary Mullen, for inspiring this project and providing guidance along the way.

Bibliography

The images describing covering arrays were taken from Dr. Moura’s webpage: http://www.site.uottawa.ca/~lucia/coveringarrays; the image depicting finite fields and Latin Squares is from the documentation for Latin Square Toolbox on https://sourceforge.net/

[1] Keedwell, A. Donald, and Gary L. Mullen. "Sets of partially orthogonal latin squares and projective planes." *Discrete Mathematics* 288.1-3 (2004): 49-60.

[2] Droz, Daniel Robert. "Orthogonal Sets of Latin Squares and Class-r Hypercubes Generated By Finite Algebraic Systems." (2016).