Computational approaches to finding pairs of latin squares with maximal orthogonality, row completeness, and diagonal completeness

Sam Auciello

01 May 2013

I. Introduction

The objective of my project was to apply computational search techniques to the problem described in *Crossover designs in the presence of carry-over effects from two factors* by Lewis and Russell.[1] A summary of this problem follows.

The goal is to find a design for a particular type of experiment that will minimize particular forms of systematic bias. For our purposes, a **design** consists of a pair of latin squares A and B of order n. A **latin square** of order n is an $n \times n$ grid with each cell filled in with one of n symbols such that each row and column of the grid contains each symbol exactly once. Here is an example latin square of order 3:

$$\begin{array}{ccc} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{array}$$

A (somewhat contrived) example of the sort of experiment that could benefit from these designs follows. We have n types of cheese, n types of wine, and n impartial judges. We want to determine which combination of one cheese and one wine the judges like best. With an unlimited supply of each wine and cheese, each judge could simply try each of the n^2 combinations. Unfortunately we might only have n servings of each cheese and wine, or our judges may only have time to try n combinations, or it may for some other reason be prohibitively expensive to have more than n^2 trials. We can set up the experiment with a series of n trials by constructing a pair of order n

latin squares A and B. In trial i judge j will try the combination of cheese $A_{i,j}$ and wine $B_{i,j}$.

By constraining both grids in the design to being latin squares, we make sure of two things. Firstly, we can be certain that each judge tries each cheese and each wine once. Secondly, we can be certain that each wine and each cheese will be tried exactly once in each time slot. This will eliminate systematic bias from, for example, always trying cheddar last or only having one judge try Merlot.

In this paper I talk about a property of multisets that I will call **redundancy**. The redundancy of a multiset is the minimum number of elements that must be removed from it to make all of its elements distinct.

A latin square A is considered **row complete** if the multiset of ordered pairs $\{(A_{i-1,j}, A_{i,j}) \mid 1 \leq i < n, 0 \leq j < n\}$ has a redundancy of 0. That is to say that each pair of adjacent symbols in the square occurs exactly once. Here is an example of a row complete latin square of order 4:

$$\begin{array}{cccc} 0 & 1 & 3 & 2 \\ 1 & 2 & 0 & 3 \\ 2 & 3 & 1 & 0 \\ 3 & 0 & 2 & 1 \end{array}$$

Note that I am using the convention of starting indices with 0 here and in the rest of this paper.

An ordered pair of latin squares A and B is considered **orthogonal** if the multiset of ordered pairs $\{(A_{i,j}, B_{i,j}) \mid 0 \leq i < n, 0 \leq j < n\}$ has a redundancy of 0. That is to say that each pair of symbols taken from the same position in each square occurs exactly once in the design. Here is an example of a pair of orthogonal latin squares of order 3:

$$\begin{array}{ccc} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{array}$$

Each design is assigned a series of five metrics M_0, M_1, M_2, M_3, M_4. The five metrics are as follows:

- M_0 is the number of places where an orthogonal pair is repeated or the redundancy of $\{(A_{i,j}, B_{i,j}) \mid 0 \leq i < n, 0 \leq j < n\}$. We want to minimize this number to maximize the number of wine/cheese combinations tried.

- M_1 is the number of mistakes in row completeness of A or the redundancy of $\{(A_{i,j-1}, A_{i,j}) \mid 0 \leq i < n, 1 \leq j < n\}$. We want to minimize this number to eliminate systematic bias that might be introduced by having the same sequence of two cheeses tried multiple times. For example, repeatedly trying cheddar just before havarti could introduce systematic bias into the evaluation of havarti.

- M_2 is the number of mistakes in row completeness of B or the redundancy of $\{(B_{i,j-1}, B_{i,j}) \mid 0 \leq i < n, 1 \leq j < n\}$. We want to minimize this number to eliminate systematic bias that might be introduced by having the same sequence of two wines tried multiple times. For example, repeatedly trying Merlot just before Pinot Noir could introduce systematic bias into the evaluation of Pinot Noir.

- M_3 works very much like M_1 and M_2 except that it looks at the pairs formed by taking one cell from A and the cell immediately to the right of it from B. For example, $(A_{1,2}, B_{1,3})$, or generally $(A_{i,j}, B_{i,j+1})$ such that $0 \leq i < n, 0 \leq j < n-1$. M_3 is the number of repeats in these pairs or the redundancy of $\{(A_{i,j-1}, B_{i,j}) \mid 0 \leq i < n, 1 \leq j < n\}$. We want to minimize this number to eliminate systematic bias from, for example, repeatedly trying cheddar just before Pinot Noir.

- M_4 works just like M_3 except in the opposite direction. It looks at the pairs formed by taking one cell from A and the cell immediately to the left of it from B. For example, $(A_{1,2}, B_{1,1})$, or generally $(A_{i,j}, B_{i,j-1})$ such that $0 \leq i < n, 1 \leq j < n$. M_4 is the number of repeats in these pairs or the redundancy of $\{(A_{i,j}, B_{i,j-1}) \mid 0 \leq i < n, 1 \leq j < n\}$. We want to minimize this number to eliminate systematic bias from, for example, repeatedly trying Merlot just before havarti.

Lewis and Russell's work was motivated by experiments in the telecommunications industry. In their experiments, what I have been thinking of as cheese was circuit conditions formed from properties of a pair of telephones and what I have been thinking of as wine was a combination of several variables associated with a transmission such as bandwidth, signal gain or loss, noise level, and coding distortion. Rather than having n judges try a series of n wine/cheese combinations, they had n pairs of subjects conduct a series of n conversations under different circuit conditions.

A common algorithm for exploring a space of possibilities is called **backtracking search**. It can be used any time that solving a problem can be represented by a series of decisions with well defined consequences. The basic structure of the recursive form of the algorithm follows:

```
Function search():
  If a solution has been found:
    Return the solution
  Otherwise:
    Initialize an empty list of solutions
    Iterate over the options at this point:
      Try this option
      If no solution can possibly be found past this point:
        Return and empty list
      Otherwise:
        Call search() and append the result to the list of
        solutions
      Undo this option
    Return the list of solutions
```

This technique can be used to find combinatorial designs by viewing the process of filling in the symbols in the design as a series of decisions between the various symbols that could occupy each space.

There is a phenomenon in combinatorics called **combinatorial explosion** where the number of designs at a given order grows very rapidly as a function of the order. For this reason, my methods use several techniques to reduce the space that must be searched. One such technique is to find symmetries that allow sections of the space to be eliminated by showing that those sections are equivalent to other squares that I am searching for. For example, permuting the names of the symbols in a square would produce another equivalent square so my search methods can often safely assume that the first row of a square is sorted in ascending order because any square I find for which this isn't true would be equivalent to one for which it is. Another technique is to keep track of the best metrics found so far and consider there to be no possible solutions beyond this point any time our intermediate metrics are worse.

In this paper I will describe several computational methods for minimizing the five metrics at various values of n. In these methods I tried first to minimize the value of $2M_0 + M_1 + M_2$, then, given the space of designs that meet that minimum, find the smallest possible value for both M_3 and M_3+M_4. The idea behind this is to give row completeness and orthogonality equal weight and then to consider diagonal completeness as secondary to that. It is considered that systematic bias from, for example, having cheddar always follow havarti is likely to be more prominent than systematic bias from having cheddar always follow Merlot, and it is easy to imagine a scenario in which having cheddar always follow Merlot is more of a concern than having Pinot Noir always follow havarti (for example, if cheeses have more of a lasting taste than wines). The statistical rationale for these metrics is more thoroughly described in Lewis and Russell.

II. Results

The following table summarizes my findings. For each value of n from 1 to 20, I've listed the lowest value I was able to find for $2M_0 + M_1 + M_2$, and among squares with that value, the lowest value for M_3 that I was able to find as well as the lowest value for $M_3 + M_4$ that I was able to find. Finally, I list the method used to find that result. In some cases, the best result for M_3 came from one method and the best result for $M_3 + M_4$ came from another. In these cases the methods are listed separately below. In some cases the same metrics were found through multiple methods. The squares themselves are listed with the description of that method with their metrics in the form $M = [M_0, M_1, M_2, M_3, M_4]$.

n	$2M_0 + M_1 + M_2$	M_3	method	$M_3 + M_4$	method
1	0	0	A	0	A
2	4	0	A	0	A
3	6	0	A	0	A
4	8	4	A	8	A
5	10	5	C	10	C
6	12	6	B,E	12	B, E
7	14	7	C	14	C
8	16	8	B, C, E	20	B
9	12	25	B	50	B
10	12	25	B	50	B
11	22	0	C	22	C
12	0	12	D, E	60	D
13	26	13	C	26	C
14	28	0	C	56	C
15	30	15	A	45	A
16	0	16	D	64	D
17	34	17	C	34	C
18	36	81	A	162	A
19	38	19	C	38	C
20	0	20	D	140	D

method key:

- A: Lewis and Russell
- B: Row permutations
- C: Directed terraces of cyclic groups
- D: Directed terraces of dihedral groups
- E: Generating arrays

The following table summarizes Lewis and Russell's results for the purpose of comparison. Numbers in **bold** have been improved upon by the methods described in this paper, numbers with no style have been matched, and numbers in *italic* were only found using Lewis and Russell's method:

n	$2M_0 + M_1 + M_2$	M_3	$M_3 + M_4$
1	0	0	0
2	4	0	0
3	6	0	0
4	8	4	8
5	10	5	**15**
6	12	**7**	**14**
7	14	**14**	**35**
8	16	**10**	20
9	**18**	18	36
10	**20**	27	54
11	22	**33**	**77**
12	**24**	**34**	**68**
13	26	**39**	**91**
14	28	**41**	**82**
15	*30*	*15*	*45*
16	**32**	**70**	**140**
17	34	**85**	**204**
18	*36*	*81*	*162*
19	38	**114**	**247**
20	**40**	**92**	**184**

III. Methods

The code for the methods listed below is available at on Github at https://github.com/olleicua/latin-squares. Documentation for the code can be found there.

Lewis and Russell

Lewis and Russell present two non-computational methods for finding reasonably good designs, one that works at even orders and another that works at odd orders.[1, section 3]

At even orders they begin by defining the two sequences $s = (0, 1, n-1, 2, n-2, 3, ...\frac{n+2}{2}, \frac{n}{2})$ and $t = (1, 3, 5, ...t-1, t, 2, 4, ...t-2)$. The pair of latin squares A and B are formed by the following formula:

$$A_{i,j} = s_j + i \bmod n$$
$$B_{i,j} = s_j + t_i \bmod n$$

This is a special case of the cyclic directed terrace method below.

Note that mod in these formulas and in the rest of this paper should be interpreted as the operation in computer science that is equivalent to divide by and take the remainder. So for example, 7 mod $3 = 1$. This operation should be thought of as returning an integer as opposed to an infinite set of integers.

At odd orders Lewis and Russell do the following. Let $m = \frac{t+1}{2}$ and $q = \lfloor \frac{m+1}{2} \rfloor$. Construct a sequence s whose even indexed elements are $(0, m+1, m+2, ...n-1, q)$ and whose odd indexed elements are $(m-1, ...q+1, q-1, ...1)$. Construct sequence t such that $t_i = s_i \times 2 \bmod n$. The pair of latin squares A and B are formed by the following formula:

$$A_{i,j} = s_j + i \bmod n$$

$$B_{i,j} = t_j + i \bmod n$$

This is a special case of the row permutations method below.

Their methods are described more thoroughly in their paper, and for the orders where I was unable to find better results, theirs are recorded here.

Results:

Order $n = 15$:

0	8	9	7	10	6	11	5	12	3	13	2	14	1	4
1	9	10	8	11	7	12	6	13	4	14	3	0	2	5
2	10	11	9	12	8	13	7	14	5	0	4	1	3	6
3	11	12	10	13	9	14	8	0	6	1	5	2	4	7
4	12	13	11	14	10	0	9	1	7	2	6	3	5	8
5	13	14	12	0	11	1	10	2	8	3	7	4	6	9
6	14	0	13	1	12	2	11	3	9	4	8	5	7	10
7	0	1	14	2	13	3	12	4	10	5	9	6	8	11
8	1	2	0	3	14	4	13	5	11	6	10	7	9	12
9	2	3	1	4	0	5	14	6	12	7	11	8	10	13
10	3	4	2	5	1	6	0	7	13	8	12	9	11	14
11	4	5	3	6	2	7	1	8	14	9	13	10	12	0
12	5	6	4	7	3	8	2	9	0	10	14	11	13	1
13	6	7	5	8	4	9	3	10	1	11	0	12	14	2
14	7	8	6	9	5	10	4	11	2	12	1	13	0	3

0	1	3	14	5	12	7	10	9	6	11	4	13	2	8
1	2	4	0	6	13	8	11	10	7	12	5	14	3	9
2	3	5	1	7	14	9	12	11	8	13	6	0	4	10
3	4	6	2	8	0	10	13	12	9	14	7	1	5	11
4	5	7	3	9	1	11	14	13	10	0	8	2	6	12
5	6	8	4	10	2	12	0	14	11	1	9	3	7	13
6	7	9	5	11	3	13	1	0	12	2	10	4	8	14
7	8	10	6	12	4	14	2	1	13	3	11	5	9	0
8	9	11	7	13	5	0	3	2	14	4	12	6	10	1
9	10	12	8	14	6	1	4	3	0	5	13	7	11	2
10	11	13	9	0	7	2	5	4	1	6	14	8	12	3
11	12	14	10	1	8	3	6	5	2	7	0	9	13	4
12	13	0	11	2	9	4	7	6	3	8	1	10	14	5
13	14	1	12	3	10	5	8	7	4	9	2	11	0	6
14	0	2	13	4	11	6	9	8	5	10	3	12	1	7

$$M = [0, 15, 15, 15, 30]$$

Order $n = 18$:

0	1	17	2	16	3	15	4	14	5	13	6	12	7	11	8	10	9
1	2	0	3	17	4	16	5	15	6	14	7	13	8	12	9	11	10
2	3	1	4	0	5	17	6	16	7	15	8	14	9	13	10	12	11
3	4	2	5	1	6	0	7	17	8	16	9	15	10	14	11	13	12
4	5	3	6	2	7	1	8	0	9	17	10	16	11	15	12	14	13
5	6	4	7	3	8	2	9	1	10	0	11	17	12	16	13	15	14
6	7	5	8	4	9	3	10	2	11	1	12	0	13	17	14	16	15
7	8	6	9	5	10	4	11	3	12	2	13	1	14	0	15	17	16
8	9	7	10	6	11	5	12	4	13	3	14	2	15	1	16	0	17
9	10	8	11	7	12	6	13	5	14	4	15	3	16	2	17	1	0
10	11	9	12	8	13	7	14	6	15	5	16	4	17	3	0	2	1
11	12	10	13	9	14	8	15	7	16	6	17	5	0	4	1	3	2
12	13	11	14	10	15	9	16	8	17	7	0	6	1	5	2	4	3
13	14	12	15	11	16	10	17	9	0	8	1	7	2	6	3	5	4
14	15	13	16	12	17	11	0	10	1	9	2	8	3	7	4	6	5
15	16	14	17	13	0	12	1	11	2	10	3	9	4	8	5	7	6
16	17	15	0	14	1	13	2	12	3	11	4	10	5	9	6	8	7
17	0	16	1	15	2	14	3	13	4	12	5	11	6	10	7	9	8

0	1	17	2	16	3	15	4	14	5	13	6	12	7	11	8	10	9
2	3	1	4	0	5	17	6	16	7	15	8	14	9	13	10	12	11
4	5	3	6	2	7	1	8	0	9	17	10	16	11	15	12	14	13
6	7	5	8	4	9	3	10	2	11	1	12	0	13	17	14	16	15
8	9	7	10	6	11	5	12	4	13	3	14	2	15	1	16	0	17
10	11	9	12	8	13	7	14	6	15	5	16	4	17	3	0	2	1
12	13	11	14	10	15	9	16	8	17	7	0	6	1	5	2	4	3
14	15	13	16	12	17	11	0	10	1	9	2	8	3	7	4	6	5
16	17	15	0	14	1	13	2	12	3	11	4	10	5	9	6	8	7
17	0	16	1	15	2	14	3	13	4	12	5	11	6	10	7	9	8
1	2	0	3	17	4	16	5	15	6	14	7	13	8	12	9	11	10
3	4	2	5	1	6	0	7	17	8	16	9	15	10	14	11	13	12
5	6	4	7	3	8	2	9	1	10	0	11	17	12	16	13	15	14
7	8	6	9	5	10	4	11	3	12	2	13	1	14	0	15	17	16
9	10	8	11	7	12	6	13	5	14	4	15	3	16	2	17	1	0
11	12	10	13	9	14	8	15	7	16	6	17	5	0	4	1	3	2
13	14	12	15	11	16	10	17	9	0	8	1	7	2	6	3	5	4
15	16	14	17	13	0	12	1	11	2	10	3	9	4	8	5	7	6

$$M = [18, 0, 0, 81, 81]$$

Row permutations

The idea behind this method is to search the space of row permutations of all squares in the space of row complete squares of a given order. To reduce the size of this space somewhat several symmetries are employed. The first step is to find the set of all row complete latin squares of order n with the following properties:

- The first row and first column are in sequential ascending order.
- If any row is swapped with the first one and then the symbols in the entire square are remapped to make the first property true again, the resulting square, if different, will come lexicographically after the original square. In this case the lexicographic ordering of squares can be produced by reading symbols down the columns from the upper left corner to the lower right corner of each square, concatenating these symbols and putting the resulting numbers in ascending order.
- If the square is reflected across the vertical axis and the symbols are remapped to make the first property true again, the resulting square, if different, will come lexicographically after the original square.

These properties assure that even though not all possible row complete squares are represented, those that are not will be equivalent to one that is. This means that the entire space of row complete latin squares of that order is being searched.

Once these squares are found, each possible pair of squares A and B is considered. For each pair, the space of permutations of the rows in B is searched for a square $B_{permuted}$ that produces the best metrics for the pair $(A, B_{permuted})$. This search process is accelerated by keeping track of the best metrics found so far and allowing the search to backtrack (by saying that no solution can be found past this point) if an intermediate result is worse then the best metrics found so far.

This means that I effectively search the space of all pairs of row complete latin squares at a given order for the pair with metrics that best match my criteria.

The space of row complete latin squares at orders 6, 8, 9, and 10 had already been found by Ian Wanless.[2] I was able to confirm his results at 6, 8, and 9. My program uses his data and completely searches the space of pairs of row complete latin squares of orders 6, 8, and 9. At order 10, the computer I was using ran out of memory before giving any output, but I was able to get some results by limiting my search to pairs of the form $(A, A_{permuted})$.

Results:
The results that I got were found in under five minutes. I was ultimately prevented from searching further by hardware limitations. The squares shown below for orders 6, 8, and 9 are the result of a complete search of the space of row complete latin squares at those orders and examining every possible pair. The order 10 results below represent about five minutes of searching the much larger space of row complete latin squares at order 10 and only comparing squares to themselves. I would estimate that this search at order ten might be possible on a better computer in a few days. Because there are 492 row complete squares of order 10 after all of the symmetry considerations,the full search that compares every possible pair could be expected to take on the order of $491! \approx 10^{1109}$ times longer. This is an excellent example of combinatorial explosion in action. At order 9 the search takes seconds; at order 10 the search takes longer than we can easily estimate.

At order $n = 6$ the row permutations method was able to surpass Lewis and Russell in M_3 and $M_3 + M_4$.

0	1	2	3	4	5	0	1	2	3	4	5
1	3	0	5	2	4	4	2	0	5	3	1
2	0	4	1	5	3	5	4	3	2	1	0
3	5	1	4	0	2	1	3	5	0	2	4
4	2	5	0	3	1	3	0	4	1	5	2
5	4	3	2	1	0	2	5	1	4	0	3

$$M = [6, 0, 0, 6, 6]$$

At order $n = 8$ the row permutations method found a design that surpasses Lewis and Russell in M_3 and another that matches them in $M_3 + M_4$.

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	6	0	7	2	4	3	5	2	4	3	5	0	7	1	6
2	0	4	6	5	7	1	3	4	6	0	2	1	3	7	5
3	7	6	4	1	0	5	2	5	2	7	6	3	1	0	4
4	2	5	1	7	3	0	6	1	7	4	0	6	2	5	3
5	4	7	0	3	6	2	1	6	5	1	4	7	0	3	2
6	3	1	5	0	2	7	4	3	0	5	7	2	6	4	1
7	5	3	2	6	1	4	0	7	3	6	1	5	4	2	0

$$M = [8, 0, 0, 8, 16]$$

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	3	0	5	2	7	4	6	1	3	0	5	2	7	4	6
2	0	4	1	6	3	7	5	4	2	6	0	7	1	5	3
3	5	1	7	0	6	2	4	5	7	3	6	1	4	0	2
4	2	6	0	7	1	5	3	7	6	5	4	3	2	1	0
5	7	3	6	1	4	0	2	6	4	7	2	5	0	3	1
6	4	7	2	5	0	3	1	3	5	1	7	0	6	2	4
7	6	5	4	3	2	1	0	2	0	4	1	6	3	7	5

$$M = [8, 0, 0, 10, 10]$$

At order $n = 9$ the row permutations method found a design that surpasses Lewis and Russell in $2M_0 + M_1 + M_2$.

$$
\begin{array}{ccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 3 & 6 & 0 & 8 & 4 & 7 & 2 & 5 \\
2 & 8 & 5 & 7 & 6 & 1 & 0 & 4 & 3 \\
3 & 0 & 7 & 1 & 5 & 8 & 2 & 6 & 4 \\
4 & 6 & 3 & 8 & 1 & 7 & 5 & 0 & 2 \\
5 & 2 & 1 & 4 & 0 & 6 & 8 & 3 & 7 \\
6 & 5 & 4 & 2 & 7 & 3 & 1 & 8 & 0 \\
7 & 4 & 8 & 6 & 2 & 0 & 3 & 5 & 1 \\
8 & 7 & 0 & 5 & 3 & 2 & 4 & 1 & 6
\end{array}
\qquad
\begin{array}{ccccccccc}
4 & 6 & 3 & 8 & 1 & 7 & 5 & 0 & 2 \\
8 & 7 & 0 & 5 & 3 & 2 & 4 & 1 & 6 \\
2 & 8 & 5 & 7 & 6 & 1 & 0 & 4 & 3 \\
5 & 2 & 1 & 4 & 0 & 6 & 8 & 3 & 7 \\
3 & 0 & 7 & 1 & 5 & 8 & 2 & 6 & 4 \\
1 & 3 & 6 & 0 & 8 & 4 & 7 & 2 & 5 \\
7 & 4 & 8 & 6 & 2 & 0 & 3 & 5 & 1 \\
6 & 5 & 4 & 2 & 7 & 3 & 1 & 8 & 0 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}
$$

$$M = [6, 0, 0, 25, 25]$$

At order $n = 10$ the row permutations method found a design that surpasses Lewis and Russell in $2M_0 + M_1 + M_2$ without exhausting the space of row complete latin squares.

$$
\begin{array}{ccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 3 & 6 & 0 & 8 & 4 & 7 & 2 & 5 \\
2 & 8 & 5 & 7 & 6 & 1 & 0 & 4 & 3 \\
3 & 0 & 7 & 1 & 5 & 8 & 2 & 6 & 4 \\
4 & 6 & 3 & 8 & 1 & 7 & 5 & 0 & 2 \\
5 & 2 & 1 & 4 & 0 & 6 & 8 & 3 & 7 \\
6 & 5 & 4 & 2 & 7 & 3 & 1 & 8 & 0 \\
7 & 4 & 8 & 6 & 2 & 0 & 3 & 5 & 1 \\
8 & 7 & 0 & 5 & 3 & 2 & 4 & 1 & 6
\end{array}
\qquad
\begin{array}{ccccccccc}
4 & 6 & 3 & 8 & 1 & 7 & 5 & 0 & 2 \\
8 & 7 & 0 & 5 & 3 & 2 & 4 & 1 & 6 \\
2 & 8 & 5 & 7 & 6 & 1 & 0 & 4 & 3 \\
5 & 2 & 1 & 4 & 0 & 6 & 8 & 3 & 7 \\
3 & 0 & 7 & 1 & 5 & 8 & 2 & 6 & 4 \\
1 & 3 & 6 & 0 & 8 & 4 & 7 & 2 & 5 \\
7 & 4 & 8 & 6 & 2 & 0 & 3 & 5 & 1 \\
6 & 5 & 4 & 2 & 7 & 3 & 1 & 8 & 0 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}
$$

$$M = [6, 0, 0, 25, 25]$$

Directed terraces of the cyclic group

This method makes use of **directed terraces**. Directed terraces are defined as follows. Let G be a group of order n. Let a be an arrangement of the elements of G. Define b to be the sequence of $n - 1$ elements such that $b_i = a_i^{-1} a_{i+1}$. If b contains every non-identity element of G then a is a directed terrace.

This method works slightly differently at even and odd orders. At even orders, a backtracking search is used to search the space of pairs of directed terraces x and y of the cyclic group of order n that minimize the number of repeats in the multiset $\{x_i - y_i \bmod n \mid 0 \leq i < n\}$. For all even orders, this minimum number of repeats has been shown to be 1.[3] My search also tries to minimize the number of repeats in the two multisets $\{x_{i-1} - y_i \bmod n \mid 1 \leq i < n\}$ and $\{x_i - y_{i-1} \bmod n \mid 1 \leq i < n\}$. It does this by keeping track of the fewest such repeats found so far and backtracking whenever the current intermediate result has more repeats.

At odd orders, the same thing is done except that instead of using a pair of directed terraces, we use a pair of sequences each of which can be constrained to only have one mistake preventing them from being a directed terrace. That is to say that $\{x_i - x_{i-1} \bmod n \mid 1 \leq i < n\}$ will contain exactly one repeat for each sequence. Additionally $\{x_i - y_i \bmod n \mid 0 \leq i < n\}$ can be constrained to contain zero repeats.

A pair of latin squares A and B can be produced from a pair of directed terraces (or sequences that approximate directed terraces) x and y using the following formulas:

$$A_{i,j} = x_j + i \bmod n$$

$$B_{i,j} = y_j + i \bmod n$$

This has the effect of propagating the repeats in the various differences that were minimized during the search through the square so that the resulting value for M_0 can be shown to be n times the number of repeats in $\{x_i - y_i \bmod n \mid 0 \leq i < n\}$ or n at even orders and 0 at odd orders. The values for M_1 and M_2 are similarly n times the number of terrace mistakes in x and y respectively or 0 at even orders and n at odd orders. The values for M_3 and M_4 will be n times the number of repeats in $\{x_{i-1} - y_i \bmod n \mid 1 \leq i < n\}$ and $\{x_i - y_{i-1} \bmod n \mid 1 \leq i < n\}$ respectively.

This method effectively searches the space of squares that can be generated in this way. Knowing that $2M_0 + M_1 + M_2$ can be constrained to 2 allows me to substantially reduce the space.
I also used a construction based on this method that produces designs with $M = [0, n, n, n, n]$ at odd prime orders to find results at orders 13, 17, and 19. The construction is to produce a pair of arrangements x and y that approximate cyclic directed terraces of order n. First choose number q such that $0 \leq q < n$. The first two elements of x are 0 then q. Each successive element of x is the previous element times q mod n. If this process produces

fewer then n elements before producing repeats then choose a new value for q and try again. There will always be a possible value of q that works.[3] y is then defined to be $(0, x_3, x_4, ...x_{n-1}, x_1, x_2)$ and the latin squares are then produced in the same way above.

Results:

The even ordered results shown below represent 63 hours, 13 minutes, and 55 seconds of running my program. I exhaustively searched the space of pairs of directed terraces of the cyclic groups of orders 2 through 14. The odd ordered results represent 35 hours, 30 minutes, and 34 seconds of running my program. I exhaustively searched the space of pairs of sequences that are one mistake away from being directed terraces of cyclic groups and produce orthogonal pairs of latin squares at orders 1 through 11.

At order $n = 5$ the cyclic directed terrace method found a design that surpasses Lewis and Russell in $M_3 + M_4$.

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

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

$$M = [0, 5, 5, 5, 5]$$

At order $n = 7$ the cyclic directed terrace method found a design that surpasses Lewis and Russell in both M_3 and $M_3 + M_4$.

0	1	3	2	6	4	5
1	2	4	3	0	5	6
2	3	5	4	1	6	0
3	4	6	5	2	0	1
4	5	0	6	3	1	2
5	6	1	0	4	2	3
6	0	2	1	5	3	4

0	2	6	4	5	1	3
1	3	0	5	6	2	4
2	4	1	6	0	3	5
3	5	2	0	1	4	6
4	6	3	1	2	5	0
5	0	4	2	3	6	1
6	1	5	3	4	0	2

$$M = [0, 7, 7, 7, 7]$$

At order $n = 8$ the cyclic directed terrace method found a design that surpasses Lewis and Russell in M_3.

0	1	7	3	6	5	2	4
1	2	0	4	7	6	3	5
2	3	1	5	0	7	4	6
3	4	2	6	1	0	5	7
4	5	3	7	2	1	6	0
5	6	4	0	3	2	7	1
6	7	5	1	4	3	0	2
7	0	6	2	5	4	1	3

0	6	1	2	7	3	5	4
1	7	2	3	0	4	6	5
2	0	3	4	1	5	7	6
3	1	4	5	2	6	0	7
4	2	5	6	3	7	1	0
5	3	6	7	4	0	2	1
6	4	7	0	5	1	3	2
7	5	0	1	6	2	4	3

$$M = [8, 0, 0, 8, 16]$$

At order $n = 11$ the cyclic directed terrace method found a designs that surpass Lewis and Russell at both M_3 and $M_3 + M_4$.

0	1	2	4	3	8	5	9	7	10	6
1	2	3	5	4	9	6	10	8	0	7
2	3	4	6	5	10	7	0	9	1	8
3	4	5	7	6	0	8	1	10	2	9
4	5	6	8	7	1	9	2	0	3	10
5	6	7	9	8	2	10	3	1	4	0
6	7	8	10	9	3	0	4	2	5	1
7	8	9	0	10	4	1	5	3	6	2
8	9	10	1	0	5	2	6	4	7	3
9	10	0	2	1	6	3	7	5	8	4
10	0	1	3	2	7	4	8	6	9	5

0	7	3	6	10	1	2	8	5	4	9
1	8	4	7	0	2	3	9	6	5	10
2	9	5	8	1	3	4	10	7	6	0
3	10	6	9	2	4	5	0	8	7	1
4	0	7	10	3	5	6	1	9	8	2
5	1	8	0	4	6	7	2	10	9	3
6	2	9	1	5	7	8	3	0	10	4
7	3	10	2	6	8	9	4	1	0	5
8	4	0	3	7	9	10	5	2	1	6
9	5	1	4	8	10	0	6	3	2	7
10	6	2	5	9	0	1	7	4	3	8

$$M = [0, 11, 11, 11, 11]$$

0	1	2	5	4	8	10	6	3	9	7
1	2	3	6	5	9	0	7	4	10	8
2	3	4	7	6	10	1	8	5	0	9
3	4	5	8	7	0	2	9	6	1	10
4	5	6	9	8	1	3	10	7	2	0
5	6	7	10	9	2	4	0	8	3	1
6	7	8	0	10	3	5	1	9	4	2
7	8	9	1	0	4	6	2	10	5	3
8	9	10	2	1	5	7	3	0	6	4
9	10	0	3	2	6	8	4	1	7	5
10	0	1	4	3	7	9	5	2	8	6

0	3	9	2	10	6	4	5	7	1	8
1	4	10	3	0	7	5	6	8	2	9
2	5	0	4	1	8	6	7	9	3	10
3	6	1	5	2	9	7	8	10	4	0
4	7	2	6	3	10	8	9	0	5	1
5	8	3	7	4	0	9	10	1	6	2
6	9	4	8	5	1	10	0	2	7	3
7	10	5	9	6	2	0	1	3	8	4
8	0	6	10	7	3	1	2	4	9	5
9	1	7	0	8	4	2	3	5	10	6
10	2	8	1	9	5	3	4	6	0	7

$$M = [0, 11, 11, 0, 33]$$

At order $n = 13$ the cyclic directed terrace method was used to construct a design that surpasses Lewis and Russell at both M_3 and $M_3 + M_4$.

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

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

$$M = [0, 13, 13, 13, 13]$$

At order $n = 14$ the cyclic directed terrace method was able to find a designs that surpass Lewis and Russell at both M_3 and $M_3 + M_4$.

0	1	4	10	3	2	13	8	12	6	11	9	5	7
1	2	5	11	4	3	0	9	13	7	12	10	6	8
2	3	6	12	5	4	1	10	0	8	13	11	7	9
3	4	7	13	6	5	2	11	1	9	0	12	8	10
4	5	8	0	7	6	3	12	2	10	1	13	9	11
5	6	9	1	8	7	4	13	3	11	2	0	10	12
6	7	10	2	9	8	5	0	4	12	3	1	11	13
7	8	11	3	10	9	6	1	5	13	4	2	12	0
8	9	12	4	11	10	7	2	6	0	5	3	13	1
9	10	13	5	12	11	8	3	7	1	6	4	0	2
10	11	0	6	13	12	9	4	8	2	7	5	1	3
11	12	1	7	0	13	10	5	9	3	8	6	2	4
12	13	2	8	1	0	11	6	10	4	9	7	3	5
13	0	3	9	2	1	12	7	11	5	10	8	4	6

0	12	2	1	6	8	9	3	13	10	5	11	4	7
1	13	3	2	7	9	10	4	0	11	6	12	5	8
2	0	4	3	8	10	11	5	1	12	7	13	6	9
3	1	5	4	9	11	12	6	2	13	8	0	7	10
4	2	6	5	10	12	13	7	3	0	9	1	8	11
5	3	7	6	11	13	0	8	4	1	10	2	9	12
6	4	8	7	12	0	1	9	5	2	11	3	10	13
7	5	9	8	13	1	2	10	6	3	12	4	11	0
8	6	10	9	0	2	3	11	7	4	13	5	12	1
9	7	11	10	1	3	4	12	8	5	0	6	13	2
10	8	12	11	2	4	5	13	9	6	1	7	0	3
11	9	13	12	3	5	6	0	10	7	2	8	1	4
12	10	0	13	4	6	7	1	11	8	3	9	2	5
13	11	1	0	5	7	8	2	12	9	4	10	3	6

$$M = [14, 0, 0, 28, 28]$$

0	1	10	6	3	9	8	13	11	4	12	2	5	7
1	2	11	7	4	10	9	0	12	5	13	3	6	8
2	3	12	8	5	11	10	1	13	6	0	4	7	9
3	4	13	9	6	12	11	2	0	7	1	5	8	10
4	5	0	10	7	13	12	3	1	8	2	6	9	11
5	6	1	11	8	0	13	4	2	9	3	7	10	12
6	7	2	12	9	1	0	5	3	10	4	8	11	13
7	8	3	13	10	2	1	6	4	11	5	9	12	0
8	9	4	0	11	3	2	7	5	12	6	10	13	1
9	10	5	1	12	4	3	8	6	13	7	11	0	2
10	11	6	2	13	5	4	9	7	0	8	12	1	3
11	12	7	3	0	6	5	10	8	1	9	13	2	4
12	13	8	4	1	7	6	11	9	2	10	0	3	5
13	0	9	5	2	8	7	12	10	3	11	1	4	6

0	13	4	11	12	1	5	2	10	6	8	3	9	7
1	0	5	12	13	2	6	3	11	7	9	4	10	8
2	1	6	13	0	3	7	4	12	8	10	5	11	9
3	2	7	0	1	4	8	5	13	9	11	6	12	10
4	3	8	1	2	5	9	6	0	10	12	7	13	11
5	4	9	2	3	6	10	7	1	11	13	8	0	12
6	5	10	3	4	7	11	8	2	12	0	9	1	13
7	6	11	4	5	8	12	9	3	13	1	10	2	0
8	7	12	5	6	9	13	10	4	0	2	11	3	1
9	8	13	6	7	10	0	11	5	1	3	12	4	2
10	9	0	7	8	11	1	12	6	2	4	13	5	3
11	10	1	8	9	12	2	13	7	3	5	0	6	4
12	11	2	9	10	13	3	0	8	4	6	1	7	5
13	12	3	10	11	0	4	1	9	5	7	2	8	6

$$M = [14, 0, 0, 0, 70]$$

At order $n = 17$ the cyclic directed terrace method was used to construct a design that surpasses Lewis and Russell at both M_3 and $M_3 + M_4$.

0	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1
1	4	10	11	14	6	16	12	0	15	9	8	5	13	3	7	2
2	5	11	12	15	7	0	13	1	16	10	9	6	14	4	8	3
3	6	12	13	16	8	1	14	2	0	11	10	7	15	5	9	4
4	7	13	14	0	9	2	15	3	1	12	11	8	16	6	10	5
5	8	14	15	1	10	3	16	4	2	13	12	9	0	7	11	6
6	9	15	16	2	11	4	0	5	3	14	13	10	1	8	12	7
7	10	16	0	3	12	5	1	6	4	15	14	11	2	9	13	8
8	11	0	1	4	13	6	2	7	5	16	15	12	3	10	14	9
9	12	1	2	5	14	7	3	8	6	0	16	13	4	11	15	10
10	13	2	3	6	15	8	4	9	7	1	0	14	5	12	16	11
11	14	3	4	7	16	9	5	10	8	2	1	15	6	13	0	12
12	15	4	5	8	0	10	6	11	9	3	2	16	7	14	1	13
13	16	5	6	9	1	11	7	12	10	4	3	0	8	15	2	14
14	0	6	7	10	2	12	8	13	11	5	4	1	9	16	3	15
15	1	7	8	11	3	13	9	14	12	6	5	2	10	0	4	16
16	2	8	9	12	4	14	10	15	13	7	6	3	11	1	5	0

0	10	13	5	15	11	16	14	8	7	4	12	2	6	1	3	9
1	11	14	6	16	12	0	15	9	8	5	13	3	7	2	4	10
2	12	15	7	0	13	1	16	10	9	6	14	4	8	3	5	11
3	13	16	8	1	14	2	0	11	10	7	15	5	9	4	6	12
4	14	0	9	2	15	3	1	12	11	8	16	6	10	5	7	13
5	15	1	10	3	16	4	2	13	12	9	0	7	11	6	8	14
6	16	2	11	4	0	5	3	14	13	10	1	8	12	7	9	15
7	0	3	12	5	1	6	4	15	14	11	2	9	13	8	10	16
8	1	4	13	6	2	7	5	16	15	12	3	10	14	9	11	0
9	2	5	14	7	3	8	6	0	16	13	4	11	15	10	12	1
10	3	6	15	8	4	9	7	1	0	14	5	12	16	11	13	2
11	4	7	16	9	5	10	8	2	1	15	6	13	0	12	14	3
12	5	8	0	10	6	11	9	3	2	16	7	14	1	13	15	4
13	6	9	1	11	7	12	10	4	3	0	8	15	2	14	16	5
14	7	10	2	12	8	13	11	5	4	1	9	16	3	15	0	6
15	8	11	3	13	9	14	12	6	5	2	10	0	4	16	1	7
16	9	12	4	14	10	15	13	7	6	3	11	1	5	0	2	8

$$M = [0, 17, 17, 17, 17]$$

At order $n = 19$ the cyclic directed terrace method was used to construct a design that surpasses Lewis and Russell at both M_3 and $M_3 + M_4$.

0	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
1	3	5	9	17	14	8	15	10	0	18	16	12	4	7	13	6	11	2
2	4	6	10	18	15	9	16	11	1	0	17	13	5	8	14	7	12	3
3	5	7	11	0	16	10	17	12	2	1	18	14	6	9	15	8	13	4
4	6	8	12	1	17	11	18	13	3	2	0	15	7	10	16	9	14	5
5	7	9	13	2	18	12	0	14	4	3	1	16	8	11	17	10	15	6
6	8	10	14	3	0	13	1	15	5	4	2	17	9	12	18	11	16	7
7	9	11	15	4	1	14	2	16	6	5	3	18	10	13	0	12	17	8
8	10	12	16	5	2	15	3	17	7	6	4	0	11	14	1	13	18	9
9	11	13	17	6	3	16	4	18	8	7	5	1	12	15	2	14	0	10
10	12	14	18	7	4	17	5	0	9	8	6	2	13	16	3	15	1	11
11	13	15	0	8	5	18	6	1	10	9	7	3	14	17	4	16	2	12
12	14	16	1	9	6	0	7	2	11	10	8	4	15	18	5	17	3	13
13	15	17	2	10	7	1	8	3	12	11	9	5	16	0	6	18	4	14
14	16	18	3	11	8	2	9	4	13	12	10	6	17	1	7	0	5	15
15	17	0	4	12	9	3	10	5	14	13	11	7	18	2	8	1	6	16
16	18	1	5	13	10	4	11	6	15	14	12	8	0	3	9	2	7	17
17	0	2	6	14	11	5	12	7	16	15	13	9	1	4	10	3	8	18
18	1	3	7	15	12	6	13	8	17	16	14	10	2	5	11	4	9	0

0	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1	2	4
1	9	17	14	8	15	10	0	18	16	12	4	7	13	6	11	2	3	5
2	10	18	15	9	16	11	1	0	17	13	5	8	14	7	12	3	4	6
3	11	0	16	10	17	12	2	1	18	14	6	9	15	8	13	4	5	7
4	12	1	17	11	18	13	3	2	0	15	7	10	16	9	14	5	6	8
5	13	2	18	12	0	14	4	3	1	16	8	11	17	10	15	6	7	9
6	14	3	0	13	1	15	5	4	2	17	9	12	18	11	16	7	8	10
7	15	4	1	14	2	16	6	5	3	18	10	13	0	12	17	8	9	11
8	16	5	2	15	3	17	7	6	4	0	11	14	1	13	18	9	10	12
9	17	6	3	16	4	18	8	7	5	1	12	15	2	14	0	10	11	13
10	18	7	4	17	5	0	9	8	6	2	13	16	3	15	1	11	12	14
11	0	8	5	18	6	1	10	9	7	3	14	17	4	16	2	12	13	15
12	1	9	6	0	7	2	11	10	8	4	15	18	5	17	3	13	14	16
13	2	10	7	1	8	3	12	11	9	5	16	0	6	18	4	14	15	17
14	3	11	8	2	9	4	13	12	10	6	17	1	7	0	5	15	16	18
15	4	12	9	3	10	5	14	13	11	7	18	2	8	1	6	16	17	0
16	5	13	10	4	11	6	15	14	12	8	0	3	9	2	7	17	18	1
17	6	14	11	5	12	7	16	15	13	9	1	4	10	3	8	18	0	2
18	7	15	12	6	13	8	17	16	14	10	2	5	11	4	9	0	1	3

$$M = [0, 19, 19, 19, 19]$$

Directed terraces of the dihedral group

This method works just like the cyclic directed terrace method except that it uses terraces based on dihedral groups instead of cyclic groups. The dihedral group of an even order n is the symmetry group of the two sided regular polyhedron with $\frac{n}{2}$ edges. The latin squares and orthogonal differences are calculated using dihedral multiplication and division rather than modular arithmetic. This means that the relevant formulas for generating a pair of full latin squares from a pair of directed terraces uses dihedral multiplication as follows:

$$A_{i,j} = x_j d_i$$
$$B_{i,j} = y_j d_i$$

where d is any arrangement of the elements of the dihedral group. With this method, M_0, M_1, and M_2 can all be constrained to zero at orders 12, 16, and 20.[4] This constraint on M_0 allowed for an additional constraint on the search space. I ran my search at these three orders.

Results:

The results below for orders 12 and 16 represent 24 hours and 17 minutes of running my program. I exhaustively searched the space of pairs of directed terraces of the dihedral group of order 12. I stopped the order 16 search early to make time for the order 20 search which ran for 61 hours, 10 minutes, and 10 seconds. The order 20 search also didn't have time to finish. I would estimate that exhaustively searching the space of pairs of dihedral terraces of order 16 could have easily taken several weeks. Exhaustively searching order 20 would have taken much longer.

At order $n = 12$ the dihedral directed terrace method found a design that surpasses Lewis and Russell on every metric.

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

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

$$M = [0, 0, 0, 12, 48]$$

At order $n = 16$ the dihedral directed terrace method found designs that surpass Lewis and Russell on every metric without searching the entire space.

0	1	2	5	7	13	6	15	4	10	14	3	11	9	12	8
1	0	15	12	10	4	11	2	13	7	3	14	6	8	5	9
2	3	4	7	9	15	8	1	6	12	0	5	13	11	14	10
3	2	1	14	12	6	13	4	15	9	5	0	8	10	7	11
4	5	6	9	11	1	10	3	8	14	2	7	15	13	0	12
5	4	3	0	14	8	15	6	1	11	7	2	10	12	9	13
6	7	8	11	13	3	12	5	10	0	4	9	1	15	2	14
7	6	5	2	0	10	1	8	3	13	9	4	12	14	11	15
8	9	10	13	15	5	14	7	12	2	6	11	3	1	4	0
9	8	7	4	2	12	3	10	5	15	11	6	14	0	13	1
10	11	12	15	1	7	0	9	14	4	8	13	5	3	6	2
11	10	9	6	4	14	5	12	7	1	13	8	0	2	15	3
12	13	14	1	3	9	2	11	0	6	10	15	7	5	8	4
13	12	11	8	6	0	7	14	9	3	15	10	2	4	1	5
14	15	0	3	5	11	4	13	2	8	12	1	9	7	10	6
15	14	13	10	8	2	9	0	11	5	1	12	4	6	3	7

0	11	12	6	14	10	1	8	5	2	3	15	13	7	9	4
1	6	5	11	3	7	0	9	12	15	14	2	4	10	8	13
2	13	14	8	0	12	3	10	7	4	5	1	15	9	11	6
3	8	7	13	5	9	2	11	14	1	0	4	6	12	10	15
4	15	0	10	2	14	5	12	9	6	7	3	1	11	13	8
5	10	9	15	7	11	4	13	0	3	2	6	8	14	12	1
6	1	2	12	4	0	7	14	11	8	9	5	3	13	15	10
7	12	11	1	9	13	6	15	2	5	4	8	10	0	14	3
8	3	4	14	6	2	9	0	13	10	11	7	5	15	1	12
9	14	13	3	11	15	8	1	4	7	6	10	12	2	0	5
10	5	6	0	8	4	11	2	15	12	13	9	7	1	3	14
11	0	15	5	13	1	10	3	6	9	8	12	14	4	2	7
12	7	8	2	10	6	13	4	1	14	15	11	9	3	5	0
13	2	1	7	15	3	12	5	8	11	10	14	0	6	4	9
14	9	10	4	12	8	15	6	3	0	1	13	11	5	7	2
15	4	3	9	1	5	14	7	10	13	12	0	2	8	6	11

$$M = [0, 0, 0, 16, 112]$$

0	1	2	10	6	15	13	8	14	9	12	3	5	11	7	4
1	0	15	7	11	2	4	9	3	8	5	14	12	6	10	13
2	3	4	12	8	1	15	10	0	11	14	5	7	13	9	6
3	2	1	9	13	4	6	11	5	10	7	0	14	8	12	15
4	5	6	14	10	3	1	12	2	13	0	7	9	15	11	8
5	4	3	11	15	6	8	13	7	12	9	2	0	10	14	1
6	7	8	0	12	5	3	14	4	15	2	9	11	1	13	10
7	6	5	13	1	8	10	15	9	14	11	4	2	12	0	3
8	9	10	2	14	7	5	0	6	1	4	11	13	3	15	12
9	8	7	15	3	10	12	1	11	0	13	6	4	14	2	5
10	11	12	4	0	9	7	2	8	3	6	13	15	5	1	14
11	10	9	1	5	12	14	3	13	2	15	8	6	0	4	7
12	13	14	6	2	11	9	4	10	5	8	15	1	7	3	0
13	12	11	3	7	14	0	5	15	4	1	10	8	2	6	9
14	15	0	8	4	13	11	6	12	7	10	1	3	9	5	2
15	14	13	5	9	0	2	7	1	6	3	12	10	4	8	11

0	14	15	12	1	6	3	7	5	11	4	13	9	10	2	8
1	3	2	5	0	11	14	10	12	6	13	4	8	7	15	9
2	0	1	14	3	8	5	9	7	13	6	15	11	12	4	10
3	5	4	7	2	13	0	12	14	8	15	6	10	9	1	11
4	2	3	0	5	10	7	11	9	15	8	1	13	14	6	12
5	7	6	9	4	15	2	14	0	10	1	8	12	11	3	13
6	4	5	2	7	12	9	13	11	1	10	3	15	0	8	14
7	9	8	11	6	1	4	0	2	12	3	10	14	13	5	15
8	6	7	4	9	14	11	15	13	3	12	5	1	2	10	0
9	11	10	13	8	3	6	2	4	14	5	12	0	15	7	1
10	8	9	6	11	0	13	1	15	5	14	7	3	4	12	2
11	13	12	15	10	5	8	4	6	0	7	14	2	1	9	3
12	10	11	8	13	2	15	3	1	7	0	9	5	6	14	4
13	15	14	1	12	7	10	6	8	2	9	0	4	3	11	5
14	12	13	10	15	4	1	5	3	9	2	11	7	8	0	6
15	1	0	3	14	9	12	8	10	4	11	2	6	5	13	7

$$M = [0, 0, 0, 32, 32]$$

At order $n = 20$ the dihedral directed terrace method found designs that surpasses Lewis and Russell on every metric without searching the entire space.

0	1	2	4	7	3	5	12	19	13	17	9	18	10	15	6	16	11	14	8
1	0	19	17	14	18	16	9	2	8	4	12	3	11	6	15	5	10	7	13
2	3	4	6	9	5	7	14	1	15	19	11	0	12	17	8	18	13	16	10
3	2	1	19	16	0	18	11	4	10	6	14	5	13	8	17	7	12	9	15
4	5	6	8	11	7	9	16	3	17	1	13	2	14	19	10	0	15	18	12
5	4	3	1	18	2	0	13	6	12	8	16	7	15	10	19	9	14	11	17
6	7	8	10	13	9	11	18	5	19	3	15	4	16	1	12	2	17	0	14
7	6	5	3	0	4	2	15	8	14	10	18	9	17	12	1	11	16	13	19
8	9	10	12	15	11	13	0	7	1	5	17	6	18	3	14	4	19	2	16
9	8	7	5	2	6	4	17	10	16	12	0	11	19	14	3	13	18	15	1
10	11	12	14	17	13	15	2	9	3	7	19	8	0	5	16	6	1	4	18
11	10	9	7	4	8	6	19	12	18	14	2	13	1	16	5	15	0	17	3
12	13	14	16	19	15	17	4	11	5	9	1	10	2	7	18	8	3	6	0
13	12	11	9	6	10	8	1	14	0	16	4	15	3	18	7	17	2	19	5
14	15	16	18	1	17	19	6	13	7	11	3	12	4	9	0	10	5	8	2
15	14	13	11	8	12	10	3	16	2	18	6	17	5	0	9	19	4	1	7
16	17	18	0	3	19	1	8	15	9	13	5	14	6	11	2	12	7	10	4
17	16	15	13	10	14	12	5	18	4	0	8	19	7	2	11	1	6	3	9
18	19	0	2	5	1	3	10	17	11	15	7	16	8	13	4	14	9	12	6
19	18	17	15	12	16	14	7	0	6	2	10	1	9	4	13	3	8	5	11

0	16	15	10	8	2	9	14	11	18	6	17	7	4	5	3	19	13	1	12
1	5	6	11	13	19	12	7	10	3	15	4	14	17	16	18	2	8	0	9
2	18	17	12	10	4	11	16	13	0	8	19	9	6	7	5	1	15	3	14
3	7	8	13	15	1	14	9	12	5	17	6	16	19	18	0	4	10	2	11
4	0	19	14	12	6	13	18	15	2	10	1	11	8	9	7	3	17	5	16
5	9	10	15	17	3	16	11	14	7	19	8	18	1	0	2	6	12	4	13
6	2	1	16	14	8	15	0	17	4	12	3	13	10	11	9	5	19	7	18
7	11	12	17	19	5	18	13	16	9	1	10	0	3	2	4	8	14	6	15
8	4	3	18	16	10	17	2	19	6	14	5	15	12	13	11	7	1	9	0
9	13	14	19	1	7	0	15	18	11	3	12	2	5	4	6	10	16	8	17
10	6	5	0	18	12	19	4	1	8	16	7	17	14	15	13	9	3	11	2
11	15	16	1	3	9	2	17	0	13	5	14	4	7	6	8	12	18	10	19
12	8	7	2	0	14	1	6	3	10	18	9	19	16	17	15	11	5	13	4
13	17	18	3	5	11	4	19	2	15	7	16	6	9	8	10	14	0	12	1
14	10	9	4	2	16	3	8	5	12	0	11	1	18	19	17	13	7	15	6
15	19	0	5	7	13	6	1	4	17	9	18	8	11	10	12	16	2	14	3
16	12	11	6	4	18	5	10	7	14	2	13	3	0	1	19	15	9	17	8
17	1	2	7	9	15	8	3	6	19	11	0	10	13	12	14	18	4	16	5
18	14	13	8	6	0	7	12	9	16	4	15	5	2	3	1	17	11	19	10
19	3	4	9	11	17	10	5	8	1	13	2	12	15	14	16	0	6	18	7

$$M = [0, 0, 0, 20, 140]$$

0	1	2	4	7	3	5	13	17	10	18	8	19	14	11	16	9	15	6	12
1	0	19	17	14	18	16	8	4	11	3	13	2	7	10	5	12	6	15	9
2	3	4	6	9	5	7	15	19	12	0	10	1	16	13	18	11	17	8	14
3	2	1	19	16	0	18	10	6	13	5	15	4	9	12	7	14	8	17	11
4	5	6	8	11	7	9	17	1	14	2	12	3	18	15	0	13	19	10	16
5	4	3	1	18	2	0	12	8	15	7	17	6	11	14	9	16	10	19	13
6	7	8	10	13	9	11	19	3	16	4	14	5	0	17	2	15	1	12	18
7	6	5	3	0	4	2	14	10	17	9	19	8	13	16	11	18	12	1	15
8	9	10	12	15	11	13	1	5	18	6	16	7	2	19	4	17	3	14	0
9	8	7	5	2	6	4	16	12	19	11	1	10	15	18	13	0	14	3	17
10	11	12	14	17	13	15	3	7	0	8	18	9	4	1	6	19	5	16	2
11	10	9	7	4	8	6	18	14	1	13	3	12	17	0	15	2	16	5	19
12	13	14	16	19	15	17	5	9	2	10	0	11	6	3	8	1	7	18	4
13	12	11	9	6	10	8	0	16	3	15	5	14	19	2	17	4	18	7	1
14	15	16	18	1	17	19	7	11	4	12	2	13	8	5	10	3	9	0	6
15	14	13	11	8	12	10	2	18	5	17	7	16	1	4	19	6	0	9	3
16	17	18	0	3	19	1	9	13	6	14	4	15	10	7	12	5	11	2	8
17	16	15	13	10	14	12	4	0	7	19	9	18	3	6	1	8	2	11	5
18	19	0	2	5	1	3	11	15	8	16	6	17	12	9	14	7	13	4	10
19	18	17	15	12	16	14	6	2	9	1	11	0	5	8	3	10	4	13	7

0	12	16	5	14	1	15	18	19	6	9	13	11	17	7	2	10	8	3	4
1	9	5	16	7	0	6	3	2	15	12	8	10	4	14	19	11	13	18	17
2	14	18	7	16	3	17	0	1	8	11	15	13	19	9	4	12	10	5	6
3	11	7	18	9	2	8	5	4	17	14	10	12	6	16	1	13	15	0	19
4	16	0	9	18	5	19	2	3	10	13	17	15	1	11	6	14	12	7	8
5	13	9	0	11	4	10	7	6	19	16	12	14	8	18	3	15	17	2	1
6	18	2	11	0	7	1	4	5	12	15	19	17	3	13	8	16	14	9	10
7	15	11	2	13	6	12	9	8	1	18	14	16	10	0	5	17	19	4	3
8	0	4	13	2	9	3	6	7	14	17	1	19	5	15	10	18	16	11	12
9	17	13	4	15	8	14	11	10	3	0	16	18	12	2	7	19	1	6	5
10	2	6	15	4	11	5	8	9	16	19	3	1	7	17	12	0	18	13	14
11	19	15	6	17	10	16	13	12	5	2	18	0	14	4	9	1	3	8	7
12	4	8	17	6	13	7	10	11	18	1	5	3	9	19	14	2	0	15	16
13	1	17	8	19	12	18	15	14	7	4	0	2	16	6	11	3	5	10	9
14	6	10	19	8	15	9	12	13	0	3	7	5	11	1	16	4	2	17	18
15	3	19	10	1	14	0	17	16	9	6	2	4	18	8	13	5	7	12	11
16	8	12	1	10	17	11	14	15	2	5	9	7	13	3	18	6	4	19	0
17	5	1	12	3	16	2	19	18	11	8	4	6	0	10	15	7	9	14	13
18	10	14	3	12	19	13	16	17	4	7	11	9	15	5	0	8	6	1	2
19	7	3	14	5	18	4	1	0	13	10	6	8	2	12	17	9	11	16	15

$$M = [0, 0, 0, 60, 80]$$

Generating Arrays

A **generating array** a grid of pairs with particular properties that allow a row complete latin square to be extrapolated from it. Specifically, a generating array R is an $r \times n$ grid of pairs (x, y) where n is divisible by r, $0 \leq x < r$, and $0 \leq y < \frac{r}{n}$. Each horizontally adjacent pair (a, b) of pairs in the array can be assigned a difference $(a_x - b_x \bmod r, a_y, b_y)$. The array should be constrained such that each possible such difference occurs exactly once and each column contains exactly one of each possible value for y. A row complete latin square A can be filled in using the formula:

$$A_{i,j} = R_{i \bmod r, j} \bmod \frac{r}{n}$$

For example consider the following 3×9 generating array:

$$\begin{array}{ccccccccc}
(0,0) & (0,1) & (1,1) & (2,0) & (1,0) & (0,2) & (2,1) & (1,2) & (2,2) \\
(0,1) & (0,2) & (0,0) & (1,1) & (2,2) & (1,0) & (1,2) & (2,1) & (2,0) \\
(0,2) & (1,0) & (2,2) & (1,2) & (1,1) & (0,1) & (2,0) & (0,0) & (2,1)
\end{array}$$

The difference between, for example, $R_{0,2}$ and $R_{0,3}$ would be calculated as $(1, 1) - (2, 0) = (1 - 2 \bmod 3, 1, 0) = (2, 1, 0)$. This generating array could be used to construct the following row complete latin square:

0	1	4	6	3	2	7	5	8
1	2	0	4	8	3	5	7	6
2	3	8	5	4	1	6	0	7
3	4	7	0	6	5	1	8	2
4	5	3	7	2	6	8	1	0
5	6	2	8	7	4	0	3	1
6	7	1	3	0	8	4	2	5
7	8	6	1	5	0	2	4	3
8	0	5	2	1	7	3	6	4

As with the directed terrace methods, repeats in pair differences for adjacent orthogonal and diagonal pairs in a pair of generating arrays are propagated through the resulting square. For example given a pair of 3×9 generating arrays R and S, if the multiset of differences

$$\{(R_{i,j,x} - S_{i,j,x} \bmod 3, R_{i,j,y}, S_{i,j,y}) \mid 0 \le i < 3, 0 \le j < 9\}$$

has a redundancy of 2 then the resulting pair of latin squares will have $M_0 = 6$.

Generating arrays can be used to construct row complete latin squares at all odd composite orders.[3]
I exhaustively searched the space of pairs of $2 \times n$ and $3 \times n$ generating arrays at orders 2 through 11 keeping track of the pair differences relevant to M_0, M_3, and M_4 and backtracking whenever the metrics were worse then a previously found result. I was also able to begin the 2×12 search.

Unfortunately, none of the results I found with this method were better than results found with other methods although it is worth noting that I was able to find some designs at order 12 with $M_0 = 0$. This means that it is possible to construct pairs of orthogonal row complete latin squares from generating arrays. With more computational resources, this could be a useful approach at orders 12, 14, and 15. Order 15 is of particular interest because Lewis and Russell's results at 15 had no row complete squares. One of the generating array pairs I found at order 12 which produces an orthogonal pair follows:

$(0,0)$	$(0,1)$	$(2,0)$	$(1,0)$	$(3,0)$	$(2,1)$	$(5,0)$	$(1,1)$	$(4,1)$	$(3,1)$	$(5,1)$	$(4,0)$
$(0,1)$	$(0,0)$	$(4,1)$	$(5,1)$	$(3,1)$	$(4,0)$	$(1,1)$	$(5,0)$	$(2,0)$	$(3,0)$	$(1,0)$	$(2,1)$

$(0,0)$	$(4,1)$	$(1,1)$	$(4,0)$	$(3,1)$	$(1,0)$	$(2,1)$	$(0,1)$	$(5,1)$	$(5,0)$	$(3,0)$	$(2,0)$
$(0,1)$	$(2,0)$	$(5,0)$	$(2,1)$	$(3,0)$	$(5,1)$	$(4,0)$	$(0,0)$	$(1,0)$	$(1,1)$	$(3,1)$	$(4,1)$

IV. Conclusion

I have looked at and implemented a variety of computational approaches to this problem. Several improvements to Lewis and Russell's results have been found. The two methods that gave the most promising results were the row permutations method and the dihedral directed terrace method. Both of these methods have somewhat limited potential to produce further useful results. The dihedral directed terrace method is limited to orders n such that $n \geq 12$ and $\frac{n}{4} \in \mathbb{Z}$.[4] The row permutations method is limited by the set of row complete latin squares it searches. At low orders it can be possible to effectively search the entire space of row complete squares. It should be noted however, that the method both Ian Wanless and myself used to reduce this space found 2 squares at order 6, 12 each at orders 8 and 9, and 492 at order 10. There are no known row complete latin squares of order 11 and the space is far to large to exhaustively search. It can only be presumed that a lot more exist at higher orders where this method would thus be far less tractable.

One way of using the row permutations method at higher orders might be to use row complete squares found using other methods. This has the disadvantage of constraining the search to squares with a particular type of structure which could severely limit the improvements that might be made. The generating array method shows particular promise here. I was able to match the $M_0 = 0$ result from the dihedral directed terrace method at order 12 which by itself suggests that the generating array method is promising. I also was able to match my results from the full row permutations method using a single square A from a generating array at order 9 (It has, been shown that all row complete squares at order 9 can be constructed from generating arrays[5, Roman-k and Tuscan-k squares p.14]) and searching the space of designs $(A, A_{permutated})$. I also tried this at order 15 but was unable to get any useful results due to the constraints of available time and hardware. There are definitely designs at order 15 from generating arrays but none that I was able to find had a low enough value for M_0 to make the charts. It is also likely that with more time and better hardware (or perhaps exploring ways of parallelizing the search) results at higher orders would be attainable using the cyclic directed terrace method.

It might also be interesting to try some searches where both M_0 and M_1 or M_2 are allowed to be greater than zero to see if this might allow $2M_0 + M_1 + M_2$ to fall below $2n$. Another interesting direction to go might be to try other search methods like simulated annealing. There are many other possibilities that could be explored and my project was only able to scratch the surface but it was a great opportunity to apply computer science to an interesting combinatorics problem with real world applications and experience combinatorial explosion first hand.

References

[1] S. M. Lewis and K.G. Russell
Crossover designs in the presence of carry-over effects from two factors
Appl. Statist. (1998)
47, Part 3, pp.379-391

[2] Data on Complete and Row-Complete Latin squares
http ://users.monash.edu.au/ iwanless/data/RCLS/
Ian Wanless,
accessed April 2013.

[3] Personal communication with Matt Ollis
2012-2013.

[4] Personal correspondence between Matt Ollis and Ian Wanless
communicated by Matt Ollis, 2012-2013.

[5] Ambrose Sterr,
Computerized mathematics and the mathematics of computation
Plan of Concentration
Marlboro College, 2007.