Investigations of Coherent Ranks Through Graph Operations on Strongly Regular Graphs

Ho Jing Rui
Supervisor: Asst Prof Gary Greaves

Abstract

The project investigates the properties of various constructions of Mutually Orthogonal Latin Squares (MOLS) and their relation to combinatorial designs. Specifically, MOLS correspond to certain 2-class association schemes, and this study explores how graph operations such as vertex switching and deletion can generalize these into coherent configurations. Using tools such as SageMath and the Weisfeiler-Lehman algorithm, the project demonstrates significant results, including the coherent rank behavior of rook graphs and the structural decomposition of adjacency matrices, revealing strong regularity. These findings contribute to the broader understanding of combinatorial objects and their symmetries.

Contents

1 Introduction

Orthogonal arrays (OAs) are fundamental in combinatorial design theory and have applications in various fields such as coding theory and experimental design. This project focuses on:

- Constructing $n - 1$ mutually orthogonal Latin squares (MOLS) of order n.

- Investigating the coherent ranks of graphs derived from orthogonal arrays and subsequently Strongly regular graphs.

- Performing graph operations (Vertex Switching, Vertex Deletion) on such graphs.

- Analysing the decomposition of adjacency matrices to explain coherent ranks.

Coherent rank and strong regularity are crucial in understanding the structure and symmetries of combinatorial objects.

2 Progress and Methodology

2.1 Constructing $n - 1$ MOLS of Order n

Initially, the project aimed to construct $n - 1$ MOLS using finite fields of size n, $\mathbb{F}_n$, where n is a prime power. This was achieved by leveraging the algebraic structure of finite fields to generate Latin squares and verify their orthogonality. More specifically, the $n - 1$ latin squares are constructed as such:

Let L_k denote the k-th Latin square, where $1 \leq k < n$. The (i, j)-th entry of L_k is defined as:

$$L_k(i, j) = a_i + a_k \cdot a_j,$$

where a_i, a_j and a_k are elements (constructors) in the finite field $\mathbb{F}_n$.

2.1.1 Why L_k is a Latin Square

A Latin square of order n is an $n \times n$ array in which each symbol from a set of n distinct elements appears exactly once in each row and each column. To show that L_k satisfies these properties, consider its construction:

$$L_k(i, j) = a_i + a_k \cdot a_j,$$

where $a_i, a_j, a_k \in \mathbb{F}_n$, the finite field of order n.

- **Row uniqueness:** Fix i and vary j. The entries in row i are given by:

$$a_i + a_k \cdot a_j \quad \text{for } j = 0, 1, \ldots, n - 1.$$

Since a_j takes all distinct values in $\mathbb{F}_n$ and a_k is a fixed non-zero element in $\mathbb{F}_n$, the values $a_k \cdot a_j$ also take all distinct values in $\mathbb{F}_n$. Adding a_i (a fixed element) to these values results in a permutation of the elements of $\mathbb{F}_n$. Thus, each row contains n distinct elements.

- **Column uniqueness:** Fix j and vary i. The entries in column j are given by:

$$a_i + a_k \cdot a_j \quad \text{for } i = 0, 1, \ldots, n-1.$$

Since a_i takes all distinct values in $\mathbb{F}_n$, the values $a_i + a_k \cdot a_j$ are also distinct for a fixed $a_k \cdot a_j$. Hence, each column contains n distinct elements.

Therefore, L_k satisfies the definition of a Latin square.

2.1.2 Why L_k and $L_{k'}$ are Mutually Orthogonal

Two Latin squares L_k and $L_{k'}$ of order n are mutually orthogonal if, when superimposed, each ordered pair of symbols appears exactly once. To verify this, consider the superposition of L_k and $L_{k'}$:

$$\big(L_k(i,j), L_{k'}(i,j)\big) = \big(a_i + a_k \cdot a_j, a_i + a_{k'} \cdot a_j\big),$$

where $k \neq k'$.

- For fixed i and j, the pair $(L_k(i,j), L_{k'}(i,j))$ is uniquely determined by the values of a_i and a_j.

- Consider two distinct pairs (i,j) and (i',j'). If $\big(L_k(i,j), L_{k'}(i,j)\big) = \big(L_k(i',j'), L_{k'}(i',j')\big)$, then:

$$a_i + a_k \cdot a_j = a_{i'} + a_k \cdot a_{j'},$$
$$a_i + a_{k'} \cdot a_j = a_{i'} + a_{k'} \cdot a_{j'}.$$

Subtracting the first equation from the second yields:

$$(a_k - a_{k'}) \cdot a_j = (a_k - a_{k'}) \cdot a_{j'}.$$

Since $a_k \neq a_{k'}$ and $\mathbb{F}_n$ is a field (implying no zero divisors), it follows that $a_j = a_{j'}$. Substituting $a_j = a_{j'}$ back into either equation implies $a_i = a_{i'}$. Thus, $(i,j) = (i',j')$.

- Therefore, each ordered pair $\big(L_k(i,j), L_{k'}(i,j)\big)$ appears exactly once, proving that L_k and $L_{k'}$ are mutually orthogonal.

2.2 Generating Graphs

The focus then shifted to analysing block graphs derived from orthogonal arrays $\mathrm{OA}(m,n)$ as well as Triangular graphs, line graphs of Complete graphs. Using the Weisfeiler-Lehman (WL) algorithm [1], coherent ranks were computed. The WL algorithm was implemented through a C++ codebase [2] integrated with SageMath.

2.2.1 Generating Orthogonal Array block graphs

We utilised the SageMath `graphs.OrthogonalArrayBlockGraph()` [3] method to instantiate the graph object. In particular, we tested for $\mathrm{OA}(2,n)$ and $\mathrm{OA}(n-1,n)$.
We later found that block graphs generated by $\mathrm{OA}(2,n)$ are isomorphic to Rook graphs of size $n \times n$ and will later use this term instead.

```
graphs.OrthogonalArrayBlockGraph(2, n).is_isomorphic(graphs.RookGraph([n,n])
# Output: True
```

2.2.2 Generating Triangular graphs

We also utilised the Sagemath `graphs.CompleteGraph().line_graph()` [4][5] method to instantiate the graph object for triangular graphs.

2.3 Deriving results from graph operations

By running the graphs through the WL algorithm, we obtained the coherent configuration matrices that allowed us to identify the fibres [6] and different coherent algebra of the matrix.

This allowed us to form the coherent closure $\mathcal{W}(L)$ and thus show the coherent rank of the matrix. The matrix of coherent configurations were also permutated according to the fibres and analysed further to identify certain patterns that held for different n. They are referred to as the Type Matrix of the graph, which have examples as shwon below.

2.4 Analysis of Rook graphs

Rook graphs, isomorphic to $OA(2, n)$, were shown to exhibit coherent ranks of:

- **10 after vertex deletion.**

- **15 after vertex switching.**

This was validated computationally and compared against theoretical predictions. The reduced type matrix are as follows:

$$\begin{bmatrix} 3 & 2 \\ & 3 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ & 3 & 2 \\ & & 3 \end{bmatrix}$$

2.5 Analysis of $OA(n-1, n)$ graphs

Block graphs constructed from orthogonal arrays of size $(n-1, n)$ also exhibit same patterns as those of rook graphs, with an exception when $n = 4$, i.e. when the Orthogonal array is of size $OA(3, 4)$. Only for this case, the coherent rank of this graph is:

- **24 after vertex deletion.**

- **31 after vertex switching.**

with the reduced type matrices as follows:

$$\begin{bmatrix} 2 & 2 & 2 \\ & 4 & 3 \\ & & 4 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 & 1 \\ & 2 & 2 & 2 \\ & & 4 & 3 \\ & & & 4 \end{bmatrix}$$

2.6 Analysis of Triangular graphs

Triangular graphs, of large size ($n > 6$), were also shown to exhibit coherent ranks of:

- **11 after vertex deletion.**

- **16 after vertex switching.**

This was validated computationally and compared against theoretical predictions. The reduced type matrix are as follows:

$$\begin{bmatrix} 3 & 2 \\ & 4 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ & 3 & 2 \\ & & 4 \end{bmatrix}$$

2.7 Decomposition of Adjacency Matrices

In order to verify the computations from the WL Algorithm, adjacency matrix decomposition was used on Rook graphs with a deleted vertex to identify the structure that leads to the explanation of its coherent rank. (Currently, this has only been done for Rook graphs and analysis for other graphs are planned to follow.)

After adjacency matrix decomposition for Rook graphs with a deleted vertex, strong relations to Strongly regular designs [7] by Sankey were observed.

3 Challenges Faced

- Learning and understanding Finite Fields without prior knowledge proved challenging yet rewarding.

- Utilising SageMath for the first time, but picked up the syntax and relevant libraries quickly.

- Interpreting the results of coherent rank calculations required significant effort, from permutating the coherent configuration matrix to scaling it into a "type matrix".

- Proving strong regularity for matrix decomposition involved theoretical work which referenced many external research papers.

4 Results and Discussion

Key findings include:

- Coherent rank of rook graphs derived from $OA(2,n)$ is 10 after deletion and 15 after vertex switching.

- Decomposition of adjacency matrices reveals a strongly regular design.

These findings align with the project objectives and demonstrate the utility of computational tools in verifying theoretical results.

5 Future Work

- Explore $OA(k, n)$ for $k > 2$ and their graph representations.

- Investigate the effect of alternative graph transformations on coherent rank.

- Formalize proofs for adjacency matrix decomposition leading to strong regularity.

- Extend applications of findings to broader combinatorial structures.

6 Conclusion

This report summarizes the progress made in investigating coherent ranks and adjacency matrix decomposition. Significant results were obtained for rook graphs and their transformations, laying the foundation for future work in combinatorial designs and graph theory.

References

[1] Neil Immerman and Rik Sengupta. The k-dimensional weisfeiler-leman algorithm. *CoRR*, abs/1907.09582, 2019.

[2] Sven Reichard. Weisfeiler-lehman stabilization code, 2025. Accessed: 2025-01-14.

[3] SageMath Developers. Orthogonal array block graph - sagemath documentation, 2025. Accessed: 2025-01-14.

[4] SageMath Developers. Complete graph - sagemath documentation, 2025. Accessed: 2025-01-14.

[5] SageMath Developers. Line graph - sagemath documentation, 2025. Accessed: 2025-01-14.

[6] Gary Greaves and Jose Yip. The coherent rank of a graph with three eigenvalues, 2024.

[7] A. D. Sankey. On strongly regular designs admitting fusion to strongly regular decomposition. *Journal of Combinatorial Designs*, 29(10), July 2021.