ON STRONGLY REGULAR GRAPHS

Majid Behbahani

A thesis
in
The Department
of
Computer Science & Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy (Computer Science)
Concordia University
Montréal, Québec, Canada

May 2009

Your file Votre référence
ISBN: 978-0-494-63369-4
Our file Notre référence
ISBN: 978-0-494-63369-4

Abstract

On strongly regular graphs

Majid Behbahani, Ph.D.
Concordia University, 2009

Strongly regular graphs are regular graphs with the additional property that the number of common neighbours for two vertices depends only on whether the vertices are adjacent or non–adjacent.

From an algebraic point of view, a graph is strongly regular if its adjacency matrix has exactly three eigenvalues. Strongly regular graphs have very interesting algebraic properties due to their strong regularity conditions.

Many strongly regular graphs are known to have large and interesting automorphism groups [23]. In [23] it is also conjectured that almost all strongly regular graphs are asymmetric. Peter Cameron in [7] mentions that "Strongly regular graphs lie on the cusp between highly structured and unstructured."

Although strongly regular graphs have been studied extensively since they were introduced, there is very little known about the automorphism group of an arbitrary strongly regular graph based on its parameters.

In this thesis, we have developed theory for studying the automorphisms of strongly regular graphs. Our study is both mathematical and computational. On the computational side, we introduce the notion of orbit matrices. Using these matrices, we were able to show that some strongly regular graphs do not admit an automorphism of a certain order.

Given the size of the automorphism, we can generate all of the orbit matrices, using a computer program. Another computer program is implemented that generates all the strongly regular graphs from that orbit matrix.

From a mathematical point of view, we have found an upper bound on the number of fixed points of the automorphisms of a strongly regular graph. This upper bound is a new upper bound and is obtained by algebraic techniques.

Acknowledgements

I am extremely grateful for the support and encouragement from my Ph.D. supervisor, Dr. Clement Lam. Without his guidance, expertise, and dedication to my work, this thesis would not have been possible. He provided me with sound advice, clear and stimulating instruction, and countless hours of his time so that I could see the completion of my thesis and Ph.D. program.

I would like to thank Dr. Clement Lam, as well as the Faculty of Engineering and Computer Science, for the support given me financially since the beginning of my program.

I would also like to express my gratitude to the other members of my thesis committee for their helpful comments and feedback: Dr. Gregory Butler, Dr. Jaroslav Opatrny, Dr. Hershy Kisilevsky, and Dr. Eric Mendelsohn.

Over the years I have been mentored by professors who helped me grow and expand my academic interests. I would like to thank Dr. Pawel Gora for his inspirational lectures on functional analysis. Thanks also to Dr. Vašek Chvátal for the thought–provoking lectures on "Discrete Mathematics of Paul Erdös", and for organising the weekly Concoco seminars. I would like to thank Dr. Hadi Kharaghani, my master's supervisor, for his continued friendship, support, and guidance.

I am grateful to the Department of Computer Science and Software Engineering staff members. I especially want to thank Halina Monkiewicz, the Graduate Programs Advisor, and Pauline Dubois, the Laboratory Coordinator. They were always available to answer my questions and assist me in any way possible.

I wish to thank my student colleagues, with whom, over many years, I had the pleasure of sharing an office: Bahman, Stephen, Azam, Tanbir, Hiba, and Narges.

I am forever indebted to my parents for their emotional and financial support, and for the sacrifices they made so that I could reach my potential. And to my wife, Evelyn, thank you for your patience, encouragement and love for these past few years.

Contents

List of Tables

List of notations

$\Gamma(\mathcal{S})$: the point graph of $\mathcal{S}$.

δ_{ij}: delta function,

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

$\delta(G)$: minimum degree of G.

ϕ: the number of fixed orbits (points).

ψ: the number of non–fixed orbits.

A: incidence matrix of a partial geometry.

B: adjacency matrix of a strongly regular graph.

$G(V, E)$: a graph G with vertex set V and edge set E.

G^c: the complement of the graph G.

$\mathrm{GQ}(s, t)$: generalised quadrangle with parameters s and t.

I: the identity matrix.

J: the all one matrix.

$\mathcal{L}$: set of lines.

$M = diag(m_1, \ldots, m_d)$.

$N = diag(n_1, \ldots, n_b)$.

$\mathrm{N}(x)$: the neighbourhood of a vertex x.

Num: number.

$\mathcal{P}$: set of points.

$\mathrm{pg}(s, t, \alpha)$: a partial geometry with parameters s, t, and α.

$\mathrm{PG}(2, n)$: a projective plane of order n.

$\mathbb{R}$: the set of real numbers.

$\mathcal{S}$: point–line structure.

$\mathcal{S}^*$: dual of point–line structure $\mathcal{S}$.

$\mathrm{srg}(v, k, \lambda, \mu)$: a strongly regular graph with parameters v, k, λ, and μ.

$T(m)$: the triangular graph of order m.

$W = B^2$

Chapter 1

Introduction and statement of the problem

In this chapter, we summarise the basic definitions and theorems about strongly regular graphs and partial geometries that we are using in the thesis. Some examples are provided for a better understanding of the theory.

1.1 Organisation of the thesis

In this thesis, we study the existence of strongly regular graphs and their automorphisms. For this purpose, we developed a computer program called the SRG program. Given an automorphism of prime order and the parameters of a strongly regular graph, the SRG program is able to tell us whether or not there is a strongly regular graph with those parameters having the given automorphism.

Strongly regular graphs have many interesting algebraic and combinatorial properties. We study the properties of strongly regular graphs in Chapter 2.

The SRG program uses the concept of orbit matrices for the generation of strongly regular graphs. In Chapter 3, we will see how orbit matrices can be generated for strongly regular graphs.

In Chapter 4. we will show how the computer program, we developed for finding strongly regular graphs, works. In this chapter, we will also show the results we obtained by running this computer program.

In chapter 5, we will show how the orbit matrices can be obtained for partial geometries. We also show how partial geometries can be generated by a computer program.

The conclusion and future work can be found in Chapter 6.

1.2 Basic definition of a strongly regular graph and an example

We start by defining the concept of a graph.

Definition 1.1 *An undirected graph G consists of a set of vertices $V(G)$, together with a set of edges $E(G)$ where an edge is an un-ordered pair of vertices.*

In this thesis, the vertex set is usually the set $\{1, 2, \ldots, v\}$. Please note that we sometimes use uv or (u, v), to represent an edge $\{u, v\}$. Two vertices u and v are *adjacent*, if $\{u, v\} \in E$, and u is a *neighbour* of v, and vice versa. Since every graph in this thesis in undirected, we shall generally omit the adjective "undirected". The number of neighbours of a vertex x is called the *degree* of x. A graph is called *regular* if all its vertices have the same degree. The minimum degree of a graph G, denoted $\delta(G)$, is the minimum degree of all the vertices of G. The *adjacency* matrix B of a graph $G(V, E)$ is the $|V| \times |V|$ matrix such that

$$B_{uv} = \begin{cases} 1 & \text{if } \{u, v\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

A *subgraph* of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

A *vertex induced* subgraph of a graph G is a subset X of vertices of G, along with all edges that have both their endpoints in X. An *edge induced* subgraph of a graph G is a subset Y of edges of G along with all vertices that are endpoints of the edges in Y.

Definition 1.2 *A strongly regular graph* $\mathrm{srg}(v,k,\lambda,\mu)$ *is a graph with* v *vertices such that the number of common neighbours of* x *and* y *is* k, λ, *or* μ *according to whether* x *and* y *are equal, adjacent, or non-adjacent, respectively.*

Example 1. Matrix B defined below is the adjacency matrix of the Petersen graph.

$$B = \left(\begin{array}{c|ccc|ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\hline
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\hline
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}\right)$$

One can verify that the Petersen graph is an $\mathrm{srg}(10,3,0,1)$ by counting for each edge, $\{u,v\}$, the number of common neighbours of u and v. For example, $\{1,2\} \notin E$ and the vertex 10 is the only common neighbour of 1 and 2, which agrees with the assertion that $\mu = 1$.

One can see that by simultaneously cyclically permuting the rows and columns in these groups, group 1 being $\{2,3,4\}$, group 2 being $\{5,6,7\}$, and group 3 being $\{8,9,10\}$, the matrix B is unchanged.

One can also see that the matrix B is subdivided into 9, 3×3 cyclic submatrices and one fixed row (column). This is a result of an automorphism of order 3 with 1 fixed point. We use the idea of automorphisms in order to reduce the size of the search in the SRG program. The search space is smaller if we know the whole matrix can be divided into cyclic submatrices. $\square$

1.3 Partial geometries and strongly regular graphs

Partial geometries are point–line structures. A point–line structure is a triple $\mathcal{S} = (\mathcal{P}, \mathcal{L}, I)$ where $\mathcal{P}$ is a set of points, $\mathcal{L}$ is a set of lines, and $I \subseteq (\mathcal{P} \times \mathcal{L}) \cup (\mathcal{L} \times \mathcal{P})$ is a symmetric incidence relation. The elements of I are also called *flags*. If $(P, l) \in I$, we say that the point P is on the line l or P and l are incident.

If $\mathcal{S}$ is a point–line structure, $\mathcal{S}^*$, the *dual* of $\mathcal{S}$, is a point–line structure such that the points (lines) of $\mathcal{S}^*$ are the lines (points) of $\mathcal{S}$. Two elements are incident in $\mathcal{S}^*$ if and only if they are incident in $\mathcal{S}$.

The *incidence* matrix A of a point–line structure is defined as follows:

$$A_{ij} = \begin{cases} 1 & \text{if point } i \text{ is on line } j \\ 0 & \text{otherwise} \end{cases}$$

Definition 1.3 *A partial linear space* $\mathrm{pls}(s, t)$ *is a point–line structure such that:*

(a) any line is incident with $s + 1$ *points, and any point with* $t + 1$ *lines;*

(b) two lines are incident with at most one point (and two points with at most one line);

If two lines are incident with a point they are called *concurrent.* If two points are incident with a line they are called *collinear.*

Definition 1.4 *A partial geometry* $\mathrm{pg}(s, t, \alpha)$ *is a* $\mathrm{pls}(s, t)$ *such that for every point* P *not incident with a line* l*, exactly* α *lines on* P *are concurrent with* l*.*

Example 2. The following example is the incidence matrix of a $\mathrm{pg}(2, 2, 1)$.

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}	l_{13}	l_{14}	l_{15}
P_1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
P_2	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
P_3	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0
P_4	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
P_5	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
P_6	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
P_7	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0
P_8	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
P_9	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
P_{10}	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
P_{11}	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
P_{12}	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0
P_{13}	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
P_{14}	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1
P_{15}	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0

The column sum in this example corresponds to the number of points on a line, which is $s+1 = 3$. The row sum corresponds to the number of lines incident on a point, which is 3 since $t = 2$. To check that it is a partial geometry, we need to verify the α–condition stated in Definition 1.4 for every point–line pair $(P, l) \notin I$. For example, we take $P = P_1$ and $l = l_2$. The set of lines incident with l_2 is $\{l_1, l_3, l_7, l_9, l_{10}, l_{12}\}$. From this set, only l_1 is incident with P_1. The α–condition can be checked similarly for the rest of the point–line pairs. □

Partial geometries and strongly regular graphs are related because the existence of a partial geometry implies the existence of two strongly regular graphs.

The *point graph* of a point–line structure $\mathcal{S}$, denoted by $\Gamma(\mathcal{S})$, is a graph where the vertices are the points of $\mathcal{S}$, and where two vertices are adjacent if the corresponding points in $\mathcal{S}$ are collinear.

For example, the matrix B defined as follows:

$$B = \left(\begin{array}{ccccc|ccccc|ccccc}
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
\hline
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
\hline
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0
\end{array}\right),$$

is the point graph of the partial geometry pg$(2, 2, 1)$, shown in Example 2. It is a strongly regular graph with parameters $v = 15$, $k = 6$, $\lambda = 1$, $\mu = 2$.

Another strongly regular graph besides the point graph can be obtained from a partial geometry. The *line graph* of a point–line structure $\mathcal{S}$ is a graph where the vertices are the lines of $\mathcal{S}$, and where two vertices are adjacent if the corresponding lines in $\mathcal{S}$ are concurrent. The line graph of a partial geometry is also strongly regular because it is the point graph of its dual.

1.4 Status of existence of strongly regular graphs and partial geometries

1.4.1 Status of existence of strongly regular graphs

One of the most important problems about strongly regular graphs is their existence. While many classes of strongly regular graphs are constructed, and there are some non–existence results, for many parameter sets the existence of a strongly regular graph is still unknown.

The first parameter set, for which we are not aware of the existence of the strongly regular graph, is $v = 65,\ k = 32,\ \lambda = 15,\ \mu = 16$.

The following table extracted from the CRC-handbook of combinatorial designs [10] shows the parameters of strongly regular graphs, with 100 or fewer vertices, whose existence are unknown. Since the complement of a strongly regular graph is

v	k	λ	μ
65	32	15	16
69	20	7	5
75	32	10	16
76	30	8	14
76	35	18	14
85	14	3	2
85	30	11	10
85	42	20	21
88	27	6	9
95	40	12	20
96	35	10	14
96	38	10	18
96	45	24	18
99	14	1	2
99	42	21	15
100	33	8	12

Table 1: Unknown strongly regular graphs with small parameters

also strongly regular (Lemma 2.3), Table 1 contains only the parameter sets with

$k < v/2$.

Our main objective within this thesis is to determine whether these unknown strongly regular graphs exist, if the order of a non–trivial automorphism is given.

1.4.2 Status of existence of partial geometries

Because of the relationship between partial geometries and strongly regular graphs, one secondary objective is to investigate the possible existence of the unknown partial geometries. An extensive amount of work has been done on the existence of partial geometries. Table 2, obtained from the CRC-handbook of combinatorial designs [10], shows the existence and non–existence of partial geometries with small parameters. It also shows the parameters of the associated point and lines graphs.

Partial Geometry				Point Graph					Line Graph				
s	t	α	Num.	v	k	λ	μ	Num.	v	k	λ	μ	Num.
2	2	1	1	15	6	1	3	1	15	6	1	3	1
2	4	1	1	27	10	1	5	1	45	12	3	3	+
3	4	2	0	28	15	6	10	4	35	16	6	8	+
3	3	1	2	40	12	2	4	28	40	12	2	4	+
4	6	3	2	45	28	15	21	1	63	30	13	15	+
3	5	1	1	64	18	2	6	167	96	20	4	4	+
5	8	4	0	66	45	28	36	1	99	48	22	24	+
6	6	4	?	70	42	23	28	+	70	42	23	28	+
4	7	2	?	75	32	10	16	?	120	35	10	10	?
3	6	1	0	76	21	2	7	0	133	24	5	4	0
5	5	2	+	81	30	9	12	+	81	30	9	12	+
4	4	1	1	85	20	3	5	+	85	20	3	5	+
6	10	5	?	91	66	45	55	1	143	70	33	35	+
4	9	2	?	95	40	12	20	?	190	45	12	10	?
5	6	2	?	96	35	10	14	?	112	36	10	12	?
5	9	3	?	96	50	22	30	?	160	54	18	18	?

Table 2: Partial geometries with small parameters

In Table 2, the column "Num." gives the exact number of non–isomorphic partial geometries, or the specific strongly regular graph if the number is known. Otherwise, a "+" denotes that one or more combinatorial object is known, and a "?" denotes that its existence is unknown.

The number with the associated point or line graph is the number of strongly regular graphs with the specified parameters, but they may not be the actual point or line graph of a partial geometry.

We note that for some unknown partial geometries, for example pg$(6, 6, 4)$ and pg$(6, 10, 5)$, candidates for their point and line graphs exist. For other cases, even the existence of candidate point and line graphs are also unknown.

1.5 Contributions

In this thesis, we developed the theory of orbit matrices for strongly regular graphs. The theory gives an efficient method to test the existence of a strongly regular graph when given an automorphism of prime order. This method, implemented using a computer program, allows us to eliminate many primes as possible divisors of the order of the automorphism group of the unknown strongly regular graphs. The remaining viable prime divisors are given in Table 3.

While testing the program, we have also found some new strongly regular graphs with parameters $v = 49$, $k = 18$, $\lambda = 7$, $\mu = 6$, that are not isomorphic to the known srg$(49, 18, 7, 6)$. In addition, we have found some new upper bounds on the number of fixed points that an automorphism of a strongly regular graph may have.

We have also generalised the theory of orbit matrices to partial geometries. As future work, a computer program can be written to expand the orbit matrix to a partial geometry.

1.6 Related work

The main result of a recent paper of Paduchikh [40] is that

G	possible primes $\{p : p \| \|Aut(G)\|\}$
srg(65, 32, 15, 16)	2,3,5
srg(69, 20, 7, 5)	2,3
srg(75, 32, 10, 16)	2,3
srg(76, 30, 8, 14)	2,3
srg(76, 35, 18, 14)	2,3,5
srg(85, 14, 3, 2)	2
srg(85, 30, 11, 10)	2,3,5,17
srg(85, 42, 20, 21)	2,3,5,7
srg(88, 27, 6, 9)	2,3,5,11
srg(95, 40, 12, 20)	2,3,5
srg(96, 35, 10, 14)	2,3,5
srg(96, 38, 10, 18)	2,3,5
srg(96, 45, 24, 18)	2,3,5
srg(99, 14, 1, 2)	2,3
srg(99, 42, 21, 15)	2,3,5,7,11
srg(100, 33, 8, 12)	2,3,5,11

Table 3: Results summarising the possible prime divisors of the order of the unknown strongly regular graphs.

Theorem 1.5 (Paduchick [40]) *If $G = \text{srg}(85, 14, 3, 2)$, ρ is an automorphism of G of prime order p, and Δ is the subgraph induced by the fixed points of ρ, then one of the following is true:*

(1) $p = 5$ or $p = 17$ and Δ is the empty graph;

(2) $p = 7$ and Δ is a 1–clique or $p = 5$ and Δ is a 5–clique;

(3) $p = 3$, Δ is a quadrangle or a 2×5 lattice, and in the last case the neighbourhoods of six vertices of Δ contain exactly two maximal cliques;

(4) $p = 2$, the neighbourhood of any vertex of Δ is connected, Δ is a union of x isolated vertices and y isolated triangles, and either $y = 1$ and $x \in \{4, 6\}$ or $y = 0$ and $x = 5$.

The proof of Theorem 1.5 is based on the character theory of the Bose–Mesner algebra of the graph.

One of the results of this thesis, as seen in Table 13, is that, the only possible prime divisor of the size of the automorphism group of srg(85, 14, 3, 2) is 2, which implies the items 1 to 3 of Theorem 1.5 are not possible.

In [35], Makhnev and Minakova, by using the same technique have shown that:

Theorem 1.6 (Makhnev, Minakova [35]) *If $G = \text{srg}(99, 14, 1, 2)$, ρ is an automorphism of G of prime order p, and Δ is the subgraph induced by the fixed points of ρ, then one of the following is true:*

(1) Δ is the singleton graph and p equals 2 or 7;

(2) Δ is the empty graph and p equals 3 or 11;

(3) Δ is the triangle graph and $p = 3$.

One of the results of this thesis, as seen in Table 21, is that, the only possible prime divisors of the size of the automorphism group of srg(85, 14, 3, 2) are 2 and 3. Moreover if $p = 3$, then there are no fixed points.

Chapter 2

Introduction to strongly regular graphs

2.1 Basic concepts

In 1963, Bose [2] introduced strongly regular graphs and partial geometries. A comprehensive survey about the construction, uniqueness, non–existence and necessary conditions for partial geometries and strongly regular graphs is given by Brouwer and van Lint [3]. We shall introduce the basic properties in this section.

The following theorem shows the relationship of the parameters of a strongly regular graph.

Theorem 2.1 *If G is an* $\mathrm{srg}(v, k, \lambda, \mu)$*, then* $k(k-\lambda-1) = \mu(v-k-1)$.

Proof To show the above equality, we count, in two different ways, the number of edges $\{y, z\}$ where $y \in \mathrm{N}(x)$ and $z \notin \mathrm{N}(x)$.

First fix point x and choose z. We have $v-k-1$ possibilities for z since $z \notin \mathrm{N}(x)$ and $z \neq x$. Now, we calculate all the possible choices for y. Any vertex that is adjacent to both x and z is a candidate for y therefore, by Definition 1.2, there are μ options for y. Thus the number of edges $\{y, z\}$ is equal to $\mu(v-k-1)$.

Next, we count the number of edges $\{y, z\}$ by choosing the vertex y first. Since

$y \in \mathrm{N}(x)$, there are k possible choices for y. Now, we calculate the number of possible choices for z. Since $z \in \mathrm{N}(y)$ and $z \notin \mathrm{N}(x)$ and $z \neq x$, there are $k-\lambda-1$ possible ways of choosing z. Therefore the number of edges $\{y, z\}$ is $k(k-\lambda-1)$ which completes the proof. □

The following lemma stems directly from Definition 1.2.

Lemma 2.2 *A symmetric* $(0,1)$*-matrix* B*, with zero on the diagonal, is the adjacency matrix of* $\mathrm{srg}(v, k, \lambda, \mu)$*, if and only if*

$$B^2 = kI + \lambda B + \mu(J - I - B). \tag{1}$$

Proof We know that B^2_{ij} is equal to the number of common neighbours of vertices i and j. Therefore by the definition of a strongly regular graph, the result follows. □

The complement of a strongly regular graph is also strongly regular.

Lemma 2.3 *The complement of an* $\mathrm{srg}(v, k, \lambda, \mu)$ *is an* $\mathrm{srg}(v, v-k, v-2k+\mu-2, v-2k+\lambda-2)$.

Proof If B is the adjacency matrix of a strongly regular graph, then $J - I - B$ is the adjacency matrix of its complement. Using Equation 1, we can see that

$$(J - I - B)^2 = (k - v)I + (v - 2k + \mu - 2)(J - I - B) + (v - 2k + \lambda - 2)B.$$

Therefore the complement of an $\mathrm{srg}(v, k, \lambda, \mu)$ is an $\mathrm{srg}(v, v - k, v - 2k + \mu - 2, v - 2k + \lambda - 2)$. □

If $G = \mathrm{srg}(v, k, \lambda, \mu)$ is disconnected, then $\mu = 0$, because there are at least two non–adjacent vertices that have no common neighbours. In this case, G is the disjoint union of m complete graphs K_n. A strongly regular graph that is connected, and its complement is connected is called *primitive* To avoid the trivial case, throughout this thesis, all strongly regular graphs are considered to be primitive.

2.2 Graph automorphism

In this thesis, we study strongly regular graphs with a non–trivial automorphism group. In this section, we review the mathematical definition of the automorphism group of a graph and its properties.

An automorphism is a permutation ρ on the vertices of a graph G such that for any $u, v \in V(G)$, $(u\rho, v\rho) \in E(G)$ if and only if $(u, v) \in E(G)$. This fact can also be expressed using the matrix notation.

If ρ is a permutation, then the corresponding *permutation matrix* $P = [p_{ij}]$ is obtained by permuting the rows of the identity matrix by permutation ρ. P can also be defined as follows:

$$p_{ij} = \begin{cases} 1 & \text{if } j\rho = i \\ 0 & \text{otherwise.} \end{cases}$$

Let A be the adjacency matrix of a graph G, a permutation matrix P is an automorphism of A if and only if

$$PAP^T = A.$$

Example 3. If

$$A = \left(\begin{array}{c|ccc|ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}\right)$$

is the adjacency matrix of a graph G, and

$$P = \left(\begin{array}{c|ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right),$$

we can see that

$$PAP^T = A.$$

Therefore P is an automorphism of G. □

Let ρ and σ be automorphisms of a graph G. Let $\rho\sigma$ denote the composition of ρ and σ. Both ρ and σ are the members of the symmetric group on $V(G)$. We have

$$\begin{aligned} xy \in E(G) &\Leftrightarrow x\rho y\rho \in E(G) \\ &\Leftrightarrow x\rho\sigma y\rho\sigma \in E(G). \end{aligned}$$

Therefore, the set of automorphisms of a graph G under composition operation is closed. Thus it forms a subgroup of the symmetric group on $V(G)$. We call this subgroup the automorphism group of G namely $\mathrm{Aut}(G)$.

The automorphism group of a graph shows us the symmetry of the graph. For more information about the automorphism group of graphs please see [1] and [6]. Frucht in [21] has shown that any group can be represented as the automorphism group of a graph. Moreover, if the group is finite, then the graph can be taken as a finite graph. Mendelsohn in [39] has further shown that for every (finite) group H, there is a (finite) strongly regular graph G such that the automorphism group of G is isomorphic to H.

If the automorphism group of a graph is the identity, then the graph is called *asymmetric*. That means the graph has no non–trivial automorphism. It is shown in [18] by Erdös that almost all graphs are asymmetric. It is also conjectured in [23] that almost all strongly regular graphs are asymmetric.

On the other hand, Peter Cameron in [7] notes that "Strongly regular graphs lie on the cusp between highly structured and unstructured".

Even though strongly regular graphs had been studied extensively since they were introduced, there is not much known about the automorphism group of an arbitrary strongly regular graph based on its parameters.

For this reason, we are interested to know more about the automorphisms of strongly regular graphs.

2.3 Algebraic properties

2.3.1 The eigenvalues of a strongly regular graph

The adjacency matrix of a strongly regular graph has interesting algebraic properties. For one, it has exactly three eigenvalues.

Let B be the adjacency matrix of a primitive $\mathrm{srg}(v, k, \lambda, \mu)$, and let $\mathbf{j}$ be an all–one vector of size v. By the definition of a strongly regular graph, the row (column) sum of B is k; thus $B\mathbf{j} = k\mathbf{j}$. Therefore, $\mathbf{j}$ is an eigenvector of B and k is one of the eigenvalues of B. We will see that the multiplicity of the eigenvalue k is 1.

We need to mention a few lemmas in order to prove the above statement. We start with the definition of a reducible matrix.

Definition 2.4 *An $n \times n$ matrix M is called* reducible *if there is a permutation matrix P such that*

$$PMP^T = \begin{pmatrix} M_1 & 0 \\ M_3 & M_2 \end{pmatrix},$$

where M_1 and M_2 are square matrices of size at least one.

A matrix is called *irreducible* if it is not reducible. Clearly the adjacency matrix of a connected graph is irreducible.

Eigenvalues of real non–negative matrices have interesting properties. The Perron–Frobenius theorem characterises the properties of real positive and non–negative matrices. We only mention the parts of the Perron-Frobenius theorem that we need. Please refer to [28] for the complete version of the theorem.

Define the *spectral radius* ρ of a matrix A to be its largest eigenvalue:

$$\rho(A) = \max\{|\lambda_i(A)|\}.$$

The proof of the following theorem is given in [28].

Theorem 2.5 (Perron–Frobenius) *Let A be a real non–negative irreducible matrix, then*

1. $\rho(A) > 0$;
2. $\rho(A)$ *is an eigenvalue of* A;
3. *The algebraic multiplicity of* $\rho(A)$ *is one.*

Another fundamental theorem that we would need to use to conclude the main result is the Geršgorin circle theorem [22].

For a square matrix $A = [a_{ij}]$, define the *deleted absolute row-sums* of A as

$$R_i = \sum_{j \neq i} |a_{ij}|.$$

The following theorem (Geršgorin circle or disc theorem) indicates that all the eigenvalues of A are inside the closed discs centred at a_{ii} with radius R_i in the complex plane $\mathbb{C}$. For the proof of this theorem, we refer the reader to [28].

Theorem 2.6 (Geršgorin) *Let $A = [a_{ij}]$ be a complex $n \times n$ matrix, and let*

$$D_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i\},$$

where R_i's are the deleted absolute row–sums, then all the eigenvalues of A lie inside the union of all D_i's.

The following lemma is related to the eigenvalues of a regular graph. For more information about eigenvalues of regular graphs please see Brualdi and Ryser [4].

Lemma 2.7 *Let A be any real irreducible non–negative matrix of order n with constant row sum k and diagonal zero. Then k is an eigenvalue of A of multiplicity equal to 1. Also if, λ is another eigenvalue of A, then $|\lambda| < k$.*

Proof Let $\mathbf{j}$ be an all one vector of size n. We have $A\mathbf{j} = k\mathbf{j}$. Therefore k is an eigenvalue of A. Since the deleted absolute row–sum of every row of A is k, and $a_{ii} = 0$ for all i, all the eigenvalues of A lie inside the disk:

$$D = \{z \in \mathbb{C} : |z| \leq k\},$$

by using Geršgorin's theorem (Theorem 2.6). Therefore no other eigenvalue of A has modulus larger than k. Since k is the largest modulus eigenvalue,

$$\rho(A) = k.$$

We assumed that A is irreducible. Thus, we can use the Perron-Frobenius theorem (Theorem 2.5), to see that the multiplicity of A is equal to 1. □

Since the strongly regular graph is primitive, it is connected, hence B is irreducible. Thus the above theorem applies and k is an eigenvalue of B with multiplicity 1. We will see that B has exactly two other eigenvalues. The following lemma is a standard result that will help us to find the other two eigenvalues.

Lemma 2.8 *Let M be any symmetric real–value matrix. Then the eigenvectors corresponding to different eigenvalues of M are orthogonal.*

Proof Assume α and β are two different eigenvalues of M and x and y are the corresponding eigenvectors. We have

$$\begin{aligned}\alpha y^T x &= y^T M x \\ &= (x^T M y)^T \\ &= (x^T \beta y)^T \\ &= \beta y^T x.\end{aligned}$$

Since $\alpha \neq \beta$, we should have $y^T x = 0$ and the previous equation is satisfied. □

Using Lemma 2.8, every other eigenvector of B should be orthogonal to $\mathbf{j}$. Let $\alpha \neq k$ be another eigenvalue of B with the corresponding eigenvector x. Applying Equation 1, we have

$$B^2 x + (\mu - \lambda) B x + (\mu - k) I x - \mu J x = 0. \tag{2}$$

Using Lemma 2.8, we have

$$Jx = 0.$$

Therefore Equation 2 simplifies to

$$\alpha^2 x + (\mu - \lambda)\alpha x + (\mu - k) x = 0.$$

Since the eigenvector $x \neq 0$, we have

$$\alpha^2 + (\mu - \lambda)\alpha + (\mu - k) = 0. \tag{3}$$

The eigenvalues of B must be the zeros of the quadratic equation (3). Therefore B has exactly two more eigenvalues, which are the solutions of Equation 3:

$$r = \frac{1}{2}\left(\lambda - \mu + \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}\right), \tag{4}$$

and

$$s = \frac{1}{2}\left(\lambda - \mu - \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}\right). \tag{5}$$

Since k is always greater than μ in a primitive strongly regular graph, the expression under the square root in Equations 4 and 5 is always positive. Therefore the two eigenvalues are always distinct.

Let f and g be the multiplicities of the eigenvalues r and s respectively. We have

$$v = 1 + f + g. \tag{6}$$

Since the sum of the eigenvalues is equal to the trace of the matrix, we have

$$k + fr + gs = tr(B) = 0. \tag{7}$$

Solving Equations 6 and 7 for f and g, and using the values of r and s in Equations 4 and 5, we find

$$f = \frac{1}{2}\left(v - 1 + \frac{(v-1)(\mu-\lambda) - 2k}{\sqrt{(\lambda-\mu)^2 + 4(k-\mu)}}\right), \tag{8}$$

and

$$g = \frac{1}{2}\left(v - 1 - \frac{(v-1)(\mu-\lambda) - 2k}{\sqrt{(\lambda-\mu)^2 + 4(k-\mu)}}\right). \tag{9}$$

Using Equations 8 and 9, and restricting f and g to non–negative integers, we have a very strong necessary condition on the parameter set (v, k, λ, μ).

If $(v-1)(\mu-\lambda)-2k \neq 0$, then the requirement that f and g be integers implies that $\sqrt{(\lambda-\mu)^2 + 4(k-\mu)}$ should be a perfect square. In this case, $\sqrt{(\lambda-\mu)^2 + 4(k-\mu)}$ is even, if and only if $\lambda - \mu$ is even. Thus, from Equations 4 and 5, the eigenvalues r and s are integers.

If $(v-1)(\mu-\lambda) - 2k = 0$, then $f = g$ and r and s need not be integers. In this case, the strongly regular graph is a *conference graph.*

The following lemma shows the eigenvalues of the complement of a strongly regular graph.

Lemma 2.9 *Let G be a strongly regular graph with parameters (v, k, λ, μ) and eigenvalues k, r, and s. Then G^c has eigenvalues $v-k-1$, $-r-1$, and $-s-1$. Moreover the eigenspaces of G^c and G are the same.*

Proof Let B be the adjacency matrix of G, then the adjacency matrix of G^c is $B^c = J - I - B$. We know that k is an eigenvalue of G and its eigenspace is the space of constant vectors. Let x be a constant vector, then, we have:

$$B^c x = (J - I - B)x = vx - x - kx = (v - k - 1)x.$$

Therefore $v - k - 1$ is an eigenvalue of G^c, and its eigenspace is the same as the eigenspace of k.

Let y be an eigenvector of B corresponding to the eigenvalue r. We know that y is orthogonal to $\mathbf{j}$. We have

$$B^c y = (J - I - B)y = 0 - y - ry = (-1 - r)y.$$

Therefore $-r-1$ is an eigenvalue of G^c, and its eigenspace is the same as the eigenspace of r.

We can show that $-1 - s$ is an eigenvalue of G^c in a similar way. □

The results of this subsection are summarised in the following theorem:

Theorem 2.10 *Let G be a strongly regular graph with parameters (v, k, λ, μ), then the eigenvalues of G have the following properties:*

1. *G has exactly three eigenvalues which are k, r, and s where*

$$r = \frac{1}{2}\left(\lambda - \mu + \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}\right),$$

and

$$s = \frac{1}{2}\left(\lambda - \mu - \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}\right).$$

2. *The multiplicity of eigenvalue k is 1 and the multiplicities of r and s are f and g respectively where*

$$f = \frac{1}{2}\left(v - 1 + \frac{(v-1)(\mu - \lambda) - 2k}{\sqrt{(\lambda - \mu)^2 + 4(k - \mu)}}\right),$$

and

$$g = \frac{1}{2}\left(v - 1 - \frac{(v-1)(\mu - \lambda) - 2k}{\sqrt{(\lambda - \mu)^2 + 4(k - \mu)}}\right).$$

3. *If $(v-1)(\mu - \lambda) - 2k \neq 0$, then the eigenvalues r and s are integers. On the other hand if $(v-1)(\mu - \lambda) - 2k = 0$, then $f = g$ and r and s need not be integers. The strongly regular graph is called a conference graph in this case.*

2.3.2 The idempotent matrices

In this subsection, we introduce the concept of idempotent matrices for strongly regular graphs. These matrices play a crucial role in the study of strongly regular graphs.

In order to study these matrices, we need to use some classical definitions and theorems of linear algebra.

The following definitions are obtained from [19] and [27], with some minor modifications. Let V be a vector space and $V_1, \ldots, V_m$ be its subspaces.

Definition 2.11 *The* sum *of* $V_1, \ldots, V_m$, *namely* $V_1 + \cdots + V_m$ *is the set of all vectors* $v_1 + \cdots + v_m$, *where* $v_i \in V_i$ *for all* $1 \leq i \leq m$.

Using the definition of vector spaces, we can see that $V_1 + \cdots + V_m$ is a subspace of V.

Definition 2.12 *A vector space* V *is said to be the* direct sum *of its subspaces* $V_1, \ldots, V_m$, *namely*

$$V = V_1 \oplus \cdots \oplus V_m$$

if and only if $V = V_1 + \cdots + V_m$ *and* $V_1, \ldots, V_m$ *are independent.*

We state the following lemma without proof.

Lemma 2.13 $V = V_1 \oplus \cdots \oplus V_m$ *if and only if every vector* $v \in V$ *has a unique representation*

$$v = v_1 + \cdots + v_m$$

for some $v_i \in V_i$.

Definition 2.14 *A linear transformation* E *is called a* projection *or* idempotent *if*

$$E^2 = E.$$

Let $V = V_1 \oplus \cdots \oplus V_m$ and let $v = v_1 + \cdots + v_m$, for $v_i \in V_i$, be the unique representation of v as mentioned in Lemma 2.13. Define $E_i v = v_i$. E_i is a linear transformation and the range of E_i is V_i. Since $E_i E_i v = E_i v_i = v_i = E_i v$, we have

$$E_i^2 = E_i.$$

Therefore E_i is a projection. The linear transformation E_i defined above is called the *projection of* V *onto* V_i.

A square matrix $A = [a_{ij}]$ with complex entries is called *Hermitian* if $a_{ij} = a_{ji}^*$. Here the superscript $*$ is the complex conjugate operation. A linear transformation is called *self-adjoint* if its matrix is Hermitian.

In this part, we explain an important theorem called the *Spectral theorem.* We use the Spectral theorem to find idempotent matrices E_1 and E_2 for any strongly regular graph.

Spectral theory has been studied extensively in operator theory. The Spectral theorem reveals the structure of normal operators on a Hilbert space. If N is a normal operator on a finite dimensional Hilbert space $\mathcal{H}$, then the theorem states that the eigenvectors of N form an orthonormal basis for $\mathcal{H}$ [12], [20].

Here, we only explain the theorem on a finite dimensional space and refer the reader to [12] for the general case. There are different ways to prove the finite dimensional case.

In order to prove the Spectral theorem, we need to use some results on self-adjoint linear transformations. We state the following theorem without a proof. For the proof of this theorem please refer to [27, page 313].

Theorem 2.15 *Let V be a real finite dimensional inner product space of positive dimension and let $T : V \to V$ be a self-adjoint linear transformation. Then T has a real non-zero eigenvector.*

The following theorem can be found in [27, page 314].

Theorem 2.16 *Let V be a finite dimensional inner product vector space over $\mathbb{R}$ and let $T : V \to V$ be a self-adjoint linear transformation. Then there exists a set of eigenvectors of T, which form an orthogonal basis for V.*

Proof Proof by induction on the dimension of V.

If $\dim V = 1$, then the proof is trivial. If $\dim V > 1$, then by Theorem 2.15, T has at least one real non–zero eigenvector x_1. Let V_1 be the subspace of V consisting of all vectors orthogonal to x_1. We have $\dim V_1 = \dim V - 1$. By the induction hypothesis V_1 has an orthogonal basis consisting of eigenvectors of T. Call this basis $\{x_2, \ldots, x_n\}$. Since $x_1.x_i = 0$, the vectors x_i are independent from x_1 for $i \geq 2$. Thus $\{x_1, \ldots, x_n\}$ forms an orthogonal basis for V. □

Theorem 2.17 (Spectral Theorem) *Let V be a finite dimensional inner product vector space over $\mathbb{R}$ and let $T : V \to V$ be a self-adjoint linear transformation. Let $\lambda_1, \ldots, \lambda_k$ be the distinct eigenvalues of T and V_i's be their associated eigenspaces. Let E_i be the projection of V on V_i. Then*

1. $V = V_1 \oplus \cdots \oplus V_k$;

2. $E_1 + \cdots + E_k = I$;

3. $T = \lambda_1 E_1 + \cdots + \lambda_k E_k$.

Proof Using Theorem 2.16, we see that $V = V_1 + \cdots + V_k$. We will show that V_i's are independent in order to prove part (1). Lemma 2.8 can be generalised to self–adjoint linear transformations using a similar proof. Therefore the eigenvectors of T corresponding to different eigenvalues are orthogonal. If $v_i \in V_i$ and $v_1 + \cdots + v_k = 0$, then, we have

$$0 = v_i.(\sum_j v_j) = \sum_j v_i v_j = ||v_i||^2.$$

Thus the subspaces V_i are independent and

$$V = V_1 \oplus \cdots \oplus V_k.$$

Let α be an arbitrary vector. Since $V = V_1 \oplus \cdots \oplus V_k$, we have $\alpha = \alpha_1 + \cdots + \alpha_k$ such that $\alpha_i \in V_i$. Since E_i's are projections of V on V_i, we have $E_i\alpha = \alpha_i$. Therefore

$$\begin{aligned}(E_1 + \cdots + E_k)\alpha &= \alpha_1 + \cdots + \alpha_k \\ &= \alpha,\end{aligned}$$

which leads to

$$E_1 + \cdots + E_k = I.$$

Let α be an arbitrary vector in V. Since E_i is the projection of V on V_i, we know that

$$\alpha_i = E_i\alpha \in V_i.$$

Since V_i is an eigenspace of T, we have

$$T\alpha_i = \lambda_i\alpha_i.$$

Yielding

$$TE_i\alpha = \lambda_i E_i\alpha$$

for any $\alpha \in V$. Hence

$$TE_i = \lambda_i E_i.$$

Since $E_1 + \cdots + E_k = I$, we have

$$\begin{aligned}T &= TI \\ &= T(E_1 + \cdots E_k) \\ &= TE_1 + \cdots TE_k \\ &= \lambda_1 E_1 + \cdots + \lambda_k E_k.\end{aligned}$$

□

A matrix U is called unitary if

$$U^*U = I.$$

The following corollary can be deduced from the Spectral theorem, but we refer the reader to [33] for its proof, since it is equivalent to the theorem we have already proved. Some textbooks call Corollary 2.18 the Spectral theorem and deduce Theorem 2.17 as a corollary [42].

Corollary 2.18 *Hermitian matrices are unitarily diagonisable which means if A is Hermitian, then*

$$A = U^* D U,$$

where the matrix U is a unitary matrix

$$D = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix};$$

and $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A.

Now, we apply the Spectral theorem to the adjacency matrix B of $G = \mathrm{srg}(v, k, \lambda, \mu)$ to find the idempotent matrices E_0, E_1, and E_2. Let V be $\mathbb{R}^v$, the v dimensional real vector space. Let V_0, V_1, and V_2 be the eigenspaces of B corresponding to the eigenvalues k, r, and s respectively. Define E_i to be the projection of V onto V_i for $i = 0, 1, 2$. These matrices are called *minimal idempotent* matrices. From Theorem 2.17, we have the following equations:

$$E_0 + E_1 + E_2 = I,$$

$$B = kE_0 + rE_1 + sE_2,$$

and

$$B^c = J - B - I = (v - k - 1)E_0 + (-r - 1)E_1 + (-s - 1)E_2.$$

Please note that the last equation comes from the fact that $J - B - I$, the adjacency matrix of the complement of G, has the same eigenspaces as B. After solving the above three sets of equations, we find:

$$E_0 = \frac{1}{v} J, \tag{10}$$

$$E_1 = \frac{1}{r-s}\left\{B - sI + \frac{s-k}{v}J\right\}, \tag{11}$$

and

$$E_2 = \frac{1}{s-r}\left\{B - rI + \frac{r-k}{v}J\right\}. \tag{12}$$

Definition 2.19 *A Hermitian $n \times n$ matrix A is called positive semidefinite if for all non–zero vectors $z \in \mathbb{C}^n$,*

$$z^*Az \geq 0. \tag{13}$$

If the inequality in Equation 13 is strict ($z^*Az > 0$), then the matrix A is called positive definite.

Lemma 2.20 *Let E be an idempotent matrix, then E is positive semidefinite.*

Proof Since E is idempotent, we have $E^2 = E$. Let x be an arbitrary complex non–zero vector and let $x = z^*E$. We have

$$(Ez)^* = x^*E^* = x^*E = x.$$

Thus

$$x^* = Ez.$$

Now, we have

$$z^*Ez = z^*E^2z = (z^*E)(Ez) = x^*x \geq 0.$$

Therefore E is positive semidefinite. □

Every principal submatrix of a positive (semi) definite matrix is also positive (semi) definite. We use this fact in our exhaustive search when the adjacency matrix of the strongly regular graph is partially discovered. The proof of the following theorem is obtained from [28] with some modification.

Theorem 2.21 *Let A be a positive semidefinite matrix of size n. Then every principal submatrix of A is positive semidefinite as well.*

Proof Let S be a subset of $\{1, 2, \dots, n\}$ and let A' be the principal submatrix of A by deleting rows and columns i from A whenever $i \in S$. Let $z = [z_i]$ be an arbitrary complex non–zero vector, such that $z_i = 0$ whenever $i \in S$.

Since A is positive semidefinte, we have

$$z^* A z \geq 0.$$

Let z' be the vector obtained by removing the i-th entries of z whenever $i \in S$. We have

$$z'^* A' z' = z^* A z \geq 0.$$

Since z' can be any arbitrary non–zero vector of $\mathbb{C}$, we conclude that A' is positive semidefinite. The proof for positive definite is similar. $\square$

Now, we can apply Lemma 2.20 to see that the matrices E_1 and E_2 in Equations 11 and 12 are positive semidefinite. Using Theorem 2.21, we can see that every principal submatrix of E_1 and E_2 is positive semidefinite.

2.4 Necessary conditions

There exists some necessary conditions on the parameter set (v, k, λ, μ). If these conditions are not satisfied, there is no srg(v, k, λ, μ). In this section, we show some of these necessary conditions.

We obtain the first necessary condition from the parameter set of the complement of $G = \text{srg}(v, k, \lambda, \mu)$. We know that $G^c = \text{srg}(v, v-k, v-2k+\mu-2, v-2k+\lambda-2)$. Therefore $v - 2k + \mu - 2 \geq 0$ and $v - 2k + \lambda - 2 \geq 0$. Thus

$$\lambda, \mu \geq 2k - v + 2. \tag{14}$$

Theorem 2.1 states that, if G is an srg(v, k, λ, μ), then

$$k(k - \lambda - 1) = \mu(v - k - 1). \tag{15}$$

which is a necessary condition on the parameters of a strongly regular graph.

Rationality conditions

In Section 2.3, we obtained Equations 8 and 9 for the multiplicities of the eigenvalues r and s of an srg(v, k, λ, μ). The fact that the multiplicities f and g should be non–negative integers together with Equations 8 and 9, places tight restrictions on the parameter set (v, k, λ, μ).

Any parameter set (v, k, λ, μ) that satisfies the rationality condition and Equations 14 and 15 is called *feasible.*

There are several other necessary conditions on the parameter set of strongly regular graphs. Among these, the most important conditions are as follows:

Kreĭn conditions

Scott in [43], using a result of M.G. Kreĭn [31], in harmonic analysis, showed that

$$(r+1)(k+r+2rs) \leq (k+r)(s+1)^2,$$

and

$$(s+1)(k+s+2rs) \leq (k+s)(r+1)^2.$$

The above two inequalities are called Kreĭn conditions. The proof of the Kreĭn conditions is long and we omit it in this thesis. We refer the reader to [46, page 237] for its proof.

It can be seen that, for example, the parameter set $v = 28,\ k = 9,\ \lambda = 0,\ \mu = 4$, is feasible, but it does not satisfy the Kreĭn conditions.

Absolute Bound

Another useful necessary condition is the so called absolute bound. Delsate, Goethals, and Seidel [16] introduced this bound, which is as follows:

$$v \leq \frac{1}{2}f(f+3).$$

The proof of absolute bound is long and we omit it in this thesis, we refer the reader to [46, page 239] for its proof.

One can check that the parameter set $v = 50,\ k = 21,\ \lambda = 4,\ \mu = 12$, is feasible and satisfies the Kreĭn conditions, but it does not satisfy the absolute bound.

2.5 Some combinatorial constructions of strongly regular graphs

In this section, we introduce some of the well–known constructions of strongly regular graphs and describe their properties.

2.5.1 Triangular graph $T(m)$

Definition 2.22 *The* triangular graph $T(m)$ *has as vertices the 2–element subsets of a set of cardinality m. Two vertices are adjacent if and only if their corresponding subsets are not disjoint.*

The triangular graph $T(m)$ can also be expressed as the line graph of the complete graph K_m.

Property 2.23 $T(m)$ *is an*

$$\mathrm{srg}(\frac{1}{2}m(m-1), 2(m-2), m-2, 4).$$

Proof Let S be a set with m elements. Since there are $\binom{m}{2}$ 2–element subsets of S, we have $v = \binom{m}{2}$. Let $A = \{a_1, a_2\}$ be a subset of S. Adjacent vertices to A are the subsets $\{a_1, x\}$ and $\{y, a_2\}$, for all $x, y \neq a_1$ and $x, y \neq a_2$. There are $m-2$ choices for x and $m-2$ choices for y, therefore $k = 2(m-1)$. Using the same method of counting, we can see that $\lambda = m-2$ and $\mu = 4$.

□

As an example, consider the complement of the Petersen graph given in Example 1. The complement of the Petersen graph is $T(5)$.

It is shown in [8] that every strongly regular graph with the same parameters as $T(m)$ is isomorphic to $T(m)$.

2.5.2 Strongly regular graphs from orthogonal arrays

Strongly regular graphs can be obtained from orthogonal arrays. An orthogonal array can be considered as a generalisation of a Latin square.

Definition 2.24 *A Latin square is a square matrix of order n such that its entries are all from the set of symbols $\{1, \ldots, n\}$ and each symbol appears exactly once in each row and exactly once in each column.*

For any integer n, one can easily find a Latin square of size n. One construction is as follows:

Take $1, 2, \ldots, n$ as the first row. Row r, $r > 1$ is obtained by a cyclic shift of the row $r - 1$ to the right.

Example 4.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \\ 2 & 3 & 4 & 1 \end{pmatrix}$$

□

Latin squares have been studied extensively. For more information on Latin squares, we refer the reader to [17] and [34].

Definition 2.25 *An orthogonal array $OA(k, n)$ is a $k \times n^2$ array of symbols from the set $\{1, \ldots, n\}$ such that for any two rows r and s, all the ordered pairs (r_i, s_i), where $1 \leq i \leq n^2$, are distinct.*

It is not difficult to see that any Latin square of order n is equivalent to an $OA(3, n)$. For example, the Latin square in Example 4 is equivalent to the following orthogonal array:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 4 & 1 & 2 & 3 & 3 & 4 & 1 & 2 & 2 & 3 & 4 & 1 \end{pmatrix}$$

Given an orthogonal array $OA(k,n)$, one can define a graph G as follows:

1. The vertices of G are the n^2 column vectors of $OA(k,n)$.

2. Two vertices are connected if and only if the corresponding vectors have the same entry in one of the coordinates.

Theorem 2.26 *The graph G defined above is an*

$$\mathrm{srg}(n^2,(n-1)k,n-2+(k-1)(k-2),k(k-1)).$$

Proof We use a simple counting method to prove the theorem. Let $X = OA(k,n)$. Two columns of X have at most one common entry in the same coordinate. First, we calculate the degree of the vertices. Let x be a vertex with the corresponding column $(x_1,\ldots,x_k)$ in X. A vertex y with the corresponding column $(y_1,\ldots,y_k)$ is adjacent to x if and only if $x_i = y_i$ for some i and $x_j \neq y_j$ for all $j \neq i$. There are $n-1$ such vertices y for each coordinate i. Therefore, the total number of vertices adjacent to x is equal to

$$k(n-1).$$

Let x and y be two adjacent vertices. Without loss of generality, we can assume that the corresponding columns of x and y are $(a,x_2,\ldots,x_k)$ and $(a,y_2,\ldots,y_k)$ respectively, where $x_i \neq y_i$ for all $2 \leq i \leq k$. Let z be a vertex adjacent to both x and y. Then the corresponding column of z is one of the following two types:

(i) $(a,z_2,\ldots,z_k)$, where $z_i \neq x_i$ and $z_i \neq y_i$ for all $2 \leq i \leq k$.

(ii) $(z_1,z_2,\ldots,z_k)$, where $z_1 \neq a$, $z_i = x_i$ and $z_j = y_j$ for only two indices i and j, and for the rest of the indices l, $z_l \neq x_l$ and $z_l \neq y_l$.

There are exactly $n-2$ columns of type (i), and exactly $(k-1)(k-2)$ columns of type (ii), therefore λ, the number of common neighbours to any two adjacent vertices in G, is equal to

$$n-2+(k-1)(k-2).$$

A method of counting similar to that used in type (ii), shows us that μ, the number of common neighbours to any two non–adjacent vertices in G is equal to

$$k(k-1).$$

$\square$

One could also obtain the strongly regular graph directly from the Latin square as follows:

Let $L = [l_{ij}]$ be a Latin square of order n. Define $A = [a_{(i,j)(i',j')}]$, $1 \leq i, j, i', j' \leq n$ be a $(0,1)$ matrix of order n^2 such that:

$$a_{(i,j)(i',j')} = \begin{cases} 1 & \text{if } i = i' \text{ or } j = j' \text{ or } l_{ij} = l_{i'j'} \\ 0 & \text{otherwise.} \end{cases}$$

Then $A - I$ is the adjacency matrix of an

$$\operatorname{srg}(n^2, 2(n-1), n, 6).$$

Example 5. Let

$$L_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \\ 2 & 3 & 4 & 1 \end{pmatrix}$$

Then the following matrix A is the adjacency matrix of the strongly regular graph G produced from L_1:

$$A = \left(\begin{array}{cccc|cccc|cccc|cccc}
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
\hline
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
\hline
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}\right).$$

It can be easily checked that the graph G is an srg$(16, 9, 4, 6)$. The complement of G is called the Shrikhande graph. It is known that there are exactly two strongly regular graphs with parameters $(16, 9, 4, 6)$ up to isomorphism. The second graph can be obtained from the following Latin square:

$$L_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

L_1 and L_2 are the only two Latin squares of order 4 up to isomorphism. $\square$

2.5.3 Strongly regular graphs obtained from a pair of skew symmetric Hadamard matrices

Here is another method of constructing strongly regular graphs. We review this method because we will use it in one of our test cases. This method was introduced by Pasechnik in [41]. In his paper he uses the concept of association schemes, but to simplify the concept, we only use adjacency matrices to show the results. By this method one could obtain a strongly regular graph from two skew symmetric Hadamard matrices of order $4n$.

Definition 2.27 *A matrix H of order n with ± 1 entries is called an Hadamard matrix if*

$$HH^T = nI,$$

where I is the identity matrix.

An Hadamard matrix H is skew symmetric if $H + H^T = 2I$.

It is not difficult to show that the order of an Hadamard matrix is either 1, 2 or $4k$. Jacques Hadamard in [24] conjectured that there exists an Hadamard matrix of order $4n$ for all integers $n > 0$. This conjecture is still open.

Example 6. H, the following matrix, is an Hadamard matrix of order 4:

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & - & - \\ 1 & - & 1 & - \\ 1 & - & - & 1 \end{pmatrix},$$

where "$-$" represents -1. □

Example 7. H, the following matrix is a skew symmetric Hadamard matrix:

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ - & 1 & 1 & - \\ - & - & 1 & 1 \\ - & 1 & - & 1 \end{pmatrix}.$$

□

A skew symmetric Hadamard matrix is called *normalised* if all the entries of the first row are positive. The matrix in Example 7, is a normalised skew symmetric Hadamard matrix.

Let H_1 and H_2 be normalised skew-symmetric Hadamard matrices of order $4n$. Let E_1 and E_2 be matrices obtained from H_1 and H_2, respectively, by removing the first row and the first column. For example, if H_1 is the matrix in Example 7, then

$$E_1 = \begin{pmatrix} 1 & 1 & - \\ - & 1 & 1 \\ 1 & - & 1 \end{pmatrix}.$$

Since the matrices H_i, $i = 1, 2$, are skew symmetric Hadamard matrices, we have

$$E_i E_i^T = 4nI - J, \tag{16}$$

and

$$E_i E_i = E_i(2I - E_i^T) = J - 4nI + 2E_i. \tag{17}$$

Moreover, let I be the identity matrix of order $4n-1$ and J be the all one matrix of order $4n-1$. Since the row sum and column sum of the E_i's are equal to 1, we have

$$E_i J = J E_i = J. \tag{18}$$

Now, we define two matrices T_1 and T_2 as follows:

$$T_i = \frac{1}{2}(E_i + J - 2I).$$

Let

$$A = T_1 \otimes T_2^T + T_1^T \otimes T_2.$$

and

$$B = T_1 \otimes T_2 + T_1^T \otimes T_2^T,$$

where the operator $\otimes$ is the matrix Kronecker product operator. Let $X = [x_{ij}]$ be an $m \times n$ matrix and Y be another matrix of any size, then

$$X \otimes Y = \begin{pmatrix} x_{11}Y & \cdots & x_{1n}Y \\ \vdots & \ddots & \vdots \\ x_{m1}Y & \cdots & x_{mn}Y \end{pmatrix}.$$

The Kronecker product has many interesting properties; however, we only mention two of them here, since we will use them later in this thesis. Let X, Y, X', and Y' be matrices of proper size, then, we have

$$XX' \otimes YY' = (X \otimes Y)(X' \otimes Y').$$

This property is called the mixed product property. We also have

$$(X \otimes Y)^T = X^T \otimes Y^T.$$

Theorem 2.28 *The matrices A and B defined above are the adjacency matrices of two strongly regular graphs with parameters*

$$((4n-1)^2, 8n^2 - 8n + 2, 4n^2 - 6n + 3, 4n^2 - 6n + 2).$$

Proof We will show that A and B satisfy Equation 22 on page 42. Before calculating A^2 and B^2, we first calculate $T_iT_i^T$ and T_i^2 since we will need to use them later. Using the definition of T_i and Equations 16 and 18, we have:

$$\begin{aligned} 4T_iT_i^T &= (E_i + J - 2I)(E_i^T + J - 2I) \\ &= E_iE_i^T + E_iJ - 2E_i + JE_i^T + J^2 - 2J - 2E_i^T - 2J + 4I \\ &= -J + 4nI + J - 2E_i + J + (4n-1)J - 2J - 2E_i^T - 2J + 4I \\ &= (4n-4)J + 4nI. \end{aligned}$$

Therefore,

$$T_iT_i^T = (n-1)J + nI. \tag{19}$$

Using Equation 17, we have:

$$\begin{aligned}
4T_i^2 &= (E+J-2I)^2 \\
&= E_i^2 + E_iJ - 2E_i + JE_i + J^2 - 2J - 2E_i - 2J + 4I^2 \\
&= E_i^2 + (4n-3)J - 4E_i + 4I \\
&= (4n-2)J - 2E_i + (4-4n)I \\
&= 4nJ - 4T_i - 4nI.
\end{aligned}$$

Therefore

$$T_i^2 = nJ - T_i - nI. \tag{20}$$

Using Equation 19, we can calculate A^2 as follows:

$$\begin{aligned}
A^2 &= (T_1 \otimes T_2^T + T_1^T \otimes T_2)^2 \\
&= T_1^2 \otimes T_2^{2^T} + T_1T_1^T \otimes T_2^TT_2 + T_1^TT_1 \otimes T_2T_2^T + T_1^{2^T} \otimes T_2^2 \\
&= [nJ - T_1 - nI] \otimes [nJ - T_2^T - nI] + [nJ - T_1^T - nI] \otimes [nJ - T_2 - nI] \\
&\quad +2[(n-1)J + nI] \otimes [(n-1)J + nI] \\
&= (2n^2 - 6n + 2)J \otimes J + (4n^2 - 2n)I \otimes I + T_1 \otimes T_2^T + T_1^T \otimes T_2 \\
&= (4n^2 - 6n + 2)J \otimes J + (4n^2 - 2n)I \otimes I + A.
\end{aligned}$$

Using Equation 20, one can calculate B^2 as well. We omit this calculation (it is similar to the calculation of A^2) and show the result which is the following:

$$B^2 = (4n^2 - 6n + 2)J \otimes J + (4n^2 - 2n)I \otimes I + A.$$

We see that both A and B satisfy Equation 22 which completes the proof. □

Example 8. Let

$$E_1 = E_2 = \begin{pmatrix} 1 & 1 & - \\ - & 1 & 1 \\ 1 & - & 1 \end{pmatrix}.$$

then

$$T_1 = T_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

After calculating the matrices A and B, we find

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

One can check that both A and B are srg$(9, 2, 1, 0)$. □

We are going to show that isomorphic normalised skew Hadamard matrices generate isomorphic strongly regular graphs. Let T_1 and T_2 be two matrices obtained from skew symmetric Hadamard matrices as mentioned above. Let T_1' be a matrix isomorphic to T_1 and T_2' be a matrix isomorphic to T_2. Let us further assume that A

and B are Pasechnic strongly regular graphs obtained from T_1 and T_2, and A' and B' are Pasechnic strongly regular graphs obtained from T_1' and T_2'.

Since T_1 is isomorphic to T_1' and T_2 is isomorphic to T_2', we have:

$$T_1' = P_1 T_1 P_1^T,$$

and

$$T_2' = P_2 T_2 P_2^T.$$

Where P_1 and P_2 are some permutation matrices, define

$$P := P_1 \otimes P_2.$$

We can see that P is a permutation matrix as well. Using the properties of the Kronecker product we mentioned on page 37, we have:

$$\begin{aligned}
A' &= T_1' \otimes T_2'^T + T_1'^T \otimes T_2' \\
&= (P_1 T_1 P_1^T) \otimes (P_2 T_2^T P_2^T) + (P_1 T_1^T P_1^T) \otimes (P_2 T_2 P_2^T) \\
&= (P_1 \otimes P_2)(T_1 \otimes T_2^T)(P_1^T \otimes P_2^T) + (P_1 \otimes P_2)(T_1^T \otimes T_2)(P_1^T \otimes P_2^T) \\
&= P(T_1 \otimes T_2^T)P^T + P(T_1^T \otimes T_2)P^T \\
&= P(T_1 \otimes T_2^T + T_1^T \otimes T_2)P^T \\
&= PAP^T.
\end{aligned}$$

As a result of the above equation, we conclude that A' is isomorphic to A. The same procedure can be followed to show that B' is isomorphic to B.

2.6 Strongly regular graphs and partial geometries

A partial geometry $\mathrm{pg}(s,t,\alpha)$ with $\alpha = 1$ is called a *generalised quadrangle* and denoted by $\mathrm{GQ}(s,t)$. A $\mathrm{GQ}(1,1)$ is the usual quadrangle with four points and four edges.

The concept of a partial geometry was introduced by Bose [2]. A partial geometry is a generalisation of the concept of a generalised quadrangle. Generalised quadrangles were introduced by Tits [45] as a generalisation of the quadrangle GQ(1,1).

Theorem 2.29 *The point graph of a* $\mathrm{pg}(s,t,\alpha)$ *is an*

$$\mathrm{SRG}(\frac{(s+1)(st+\alpha)}{\alpha}, s(t+1), s-1+t(\alpha-1), \alpha(t+1)). \tag{21}$$

Proof Since there are $t+1$ lines passing through a point P and there are s points other than P on each of these lines, we have $k = s(t+1)$. Let P_1 and P_2 be two points on a line l. There are $s-1$ other points on l which all are collinear with both P_1 and P_2. Now, we count the number of points not on l which are collinear with both P_1 and P_2. There are t lines $(l_1, l_2, \ldots, l_t)$ passing through P_1 other than l. Exactly α lines on P_2 are concurrent with each l_i, $i = 1, 2, \ldots, t$, (one of these lines is l, therefore $\alpha - 1$ points on l_i are collinear with both P_1 and P_2 and are not on l). Therefore, we have $t(\alpha-1)$ points not on l, collinear with both P_1 and P_2. Thus $\lambda = s-1+t(\alpha-1)$.

Using a similar argument, we can see that $\mu = \alpha(t+1)$. The value of v can be obtained using Theorem 2.1 □

A strongly regular graph of the form (21) is called a *psuedogeometric* (s,t,α)*-graph*. Such a graph is geometric if it is the point graph of a $\mathrm{pg}(s,t,\alpha)$.

Lemma 2.30 *A psuedogeometric graph is geometric if and only if it is the point graph of a partial linear space.*

Proof The following proof is based on [9].

Since any partial geometry is also a partial linear space, the necessary condition is easily resolved. We shall now prove the sufficient condition.

Let $\mathcal{S} = \mathrm{pls}(s,t)$. Because the point graph is geometric, we have

$$\Gamma(\mathcal{S}) = \mathrm{srg}(\frac{(s+1)(st+\alpha)}{\alpha}, s(t+1), s-1+t(\alpha-1), \alpha(t+1)),$$

for some integer α. It is enough to show that $\mathcal{S} = \mathrm{pg}(s,t,\alpha)$. Fix a line l. For a point P not incident with l, let α_P be the number of lines on P concurrent with l.

We have:

$$\begin{aligned}\sum_{P\notin l}\alpha_P &= \sum_{M\in l}|\{Q\notin l, M\sim Q\}| \\ &= \sum_{M\in l} ts \\ &= (s+1)ts.\end{aligned}$$

We also have:

$$\begin{aligned}\sum_{P\notin l}\binom{\alpha_P}{2} &= \sum_{M,N\in l}|\{Q\notin l, M\sim Q, N\sim Q\}| \\ &= \sum_{M,N\in l}(\lambda-(s-1)) \\ &= \binom{s+1}{2}t(\alpha-1).\end{aligned}$$

Therefore:

$$\begin{aligned}\sum_{P\notin l}(\alpha-\alpha_P)^2 &= \sum_{P\notin l}\alpha^2+\sum_{P\notin l}(\alpha_P^2-\alpha_P)-(2\alpha-1)\sum_{P\notin l}\alpha_P \\ &= \sum_{P\notin l}\alpha^2+2\sum_{P\notin l}\binom{\alpha_P}{2}-(2\alpha-1)\sum_{P\notin l}\alpha_P \\ &= 0.\end{aligned}$$

Thus $\alpha_P=\alpha$ for every $P\notin l$ and $\mathcal{S}=\mathrm{pg}(s,t,\alpha)$. □

The adjacency matrix of the point graph can be obtained easily from the incidence matrix of the partial geometry. Let A be the incidence matrix of $\mathcal{P}=\mathrm{pg}(s,t,\alpha)$ and B be the adjacency matrix of the point graph of $\mathcal{P}$. The (i,j) entry of B^2 is the number of vertices adjacent to i and j. Since the point graph is an $\mathrm{srg}(v,k,\lambda,\mu)$, we have

$$(B^2)_{ij}=\begin{cases}k & \text{if } i=j\\ \lambda & \text{if } B_{ij}=1\\ \mu & \text{if } B_{ij}=0.\end{cases}$$

Thus

$$B^2=(k-\mu)I+\mu J+(\lambda-\mu)B. \tag{22}$$

Since A is the adjacency matrix of $\mathcal{P}$, we have

$$(AA^T)_{ij} = \begin{cases} 1 & \text{if the points } i \text{ and } j \text{ are collinear} \\ 0 & \text{if the points } i \text{ and } j \text{ are not collinear} \\ t+1 & \text{if } i = j. \end{cases}$$

Therefore by the definition of a point graph,

$$AA^T = B + (t+1)I. \tag{23}$$

Constructing a partial geometry from its point graph has been unsuccessful in most of the cases, but Haemers in [25] constructed a $\mathrm{pg}(4, 17, 2)$ from an $\mathrm{srg}(175, 72, 20, 36)$.

Chapter 3

Orbit matrices

3.1 Introduction

The size of the search for the unknown strongly regular graphs we are interested in is very large. We have to use mathematical techniques to reduce this size. One of the techniques is the use of an automorphism group. In this chapter, we shall show how the assumption of a non–trivial automorphism group may enable us to finish the search in a feasible amount of time.

Assuming that the strongly regular graph has a non–trivial automorphism group, in this chapter we develop the theory of orbit matrices for strongly regular graph for the first time.

3.2 Related work

Rudolf Mathon in [37] introduced the concept of orbit matrices for block designs. In that paper, orbit matrices are referred to as "tactical decompositions". Clement Lam in [32] showed how to use orbit matrices by the use of a program called BDX to construct block designs. Rudolf Mathon in [36] introduced the concept of *block valencies* for self–complementary strongly regular graphs. A graph is called self–complementary if there is a permutation ρ on its vertices which maps every edge to a non–edge and vice versa. Block valencies are computed based on the partitions that ρ induces on the adjacency matrix of the self–complementary strongly regular graph. Using these, a computer program was implemented in [36] to find all self complementary strongly regular graphs with less that 54 vertices.

3.3 The construction

Let $G(V, E)$ be an srg(v, k, λ, μ). Suppose an automorphism of G partitions the set of vertices V into b orbits $O_1, O_2, \ldots, O_b$. Define $n_i = |O_i|$ for $1 \leq i \leq b$.

Let $v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G that preserves the ordering $(O_1, O_2, \ldots, O_b)$. In other words, if $i < j$, then for all $v_l \in O_i$ and $v_m \in O_j$, $l < m$.

Using this ordering, the orbits of V divides the adjacency matrix B of G into submatrices

$$B = [B_{ij}],$$

where B_{ij} is the adjacency matrix of vertices in O_i versus vertices in O_j.

As an example, consider the point graph of the partial geometry pg$(2, 2, 1)$ shown in Example 2 on page 4. It is an srg$(15, 6, 1, 2)$. As a point graph, the vertices are the points $\{P_1, P_2, \ldots, P_{15}\}$. For simplicity, we denote vertex P_i by i. The permutation $(1, 2, 3, 4, 5)(6, 7, 8, 9, 10)(11, 12, 13, 14, 15)$ on the vertices is an automorphism of the point graph. This automorphism produces three orbits of size five, namely $O_1 = \{1, 2, 3, 4, 5\}$, $O_2 = \{6, 7, 8, 9, 10\}$, and $O_3 = \{11, 12, 13, 14, 15\}$.

$$
B = \left(\begin{array}{ccccc|ccccc|ccccc}
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
\hline
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
\hline
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0
\end{array}\right)
$$

The row sum and column sum of the submatrices B_{ij} are extremely important in the theory we are developing. Define three matrices $C = [c_{ij}]$, $R = [r_{ij}]$, $1 \leq i, j \leq b$, and N such that

$$c_{ij} = \text{column sum of } B_{ij},$$

$$r_{ij} = \text{row sum of } B_{ij},$$

and

$$N = \operatorname{diag}(n_1, n_2, \ldots n_b).$$

Note that R is related to C by the formula

$$r_{ij} = c_{ij}\left(\frac{n_j}{n_i}\right).$$

Since the adjacency matrix is symmetric,

$$R = C^T.$$

For example, the matrices C and R corresponding to the matrix B are:

$$C = \begin{pmatrix} 2 & 3 & 1 \\ 3 & 0 & 3 \\ 1 & 3 & 2 \end{pmatrix},$$

and

$$R = \begin{pmatrix} 2 & 3 & 1 \\ 3 & 0 & 3 \\ 1 & 3 & 2 \end{pmatrix}.$$

The orbit sizes are on the diagonal of the matrix N

$$N = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}.$$

The matrix C is the *orbit matrix* of the graph G. It gives structural information about the adjacency matrix B.

We next establish a relationship involving the matrices C, N, and R. This will be our starting point for a computer enumeration of all the possible orbit matrices C for a given orbit partition of the vertices.

Let $W = B^2$, where W_{uv} counts the number of paths of lengths 2 between vertices u and v. Using the same orbit partition of the vertices, W can also be partitioned in $W = [W_{ij}]$, where i and j are the indices of the orbits.

We continue our example of srg$(15, 6, 1, 2)$ and calculate W as follows:

$$W = B^2 = \left(\begin{array}{ccccc|ccccc|ccccc}
6 & 1 & 3 & 3 & 1 & 1 & 3 & 1 & 3 & 1 & 3 & 3 & 1 & 3 & 3 \\
1 & 6 & 1 & 3 & 3 & 1 & 1 & 3 & 1 & 3 & 3 & 3 & 3 & 1 & 3 \\
3 & 1 & 6 & 1 & 3 & 3 & 1 & 1 & 3 & 1 & 3 & 3 & 3 & 3 & 1 \\
3 & 3 & 1 & 6 & 1 & 1 & 3 & 1 & 1 & 3 & 1 & 3 & 3 & 3 & 3 \\
1 & 3 & 3 & 1 & 6 & 3 & 1 & 3 & 1 & 1 & 3 & 1 & 3 & 3 & 3 \\
\hline
1 & 1 & 3 & 1 & 3 & 6 & 3 & 3 & 3 & 3 & 1 & 1 & 3 & 3 & 1 \\
3 & 1 & 1 & 3 & 1 & 3 & 6 & 3 & 3 & 3 & 1 & 1 & 1 & 3 & 3 \\
1 & 3 & 1 & 1 & 3 & 3 & 3 & 6 & 3 & 3 & 3 & 1 & 1 & 1 & 3 \\
3 & 1 & 3 & 1 & 1 & 3 & 3 & 3 & 6 & 3 & 3 & 3 & 1 & 1 & 1 \\
1 & 3 & 1 & 3 & 1 & 3 & 3 & 3 & 3 & 6 & 1 & 3 & 3 & 1 & 1 \\
\hline
3 & 3 & 3 & 1 & 3 & 1 & 1 & 3 & 3 & 1 & 6 & 3 & 1 & 1 & 3 \\
3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 6 & 3 & 1 & 1 \\
1 & 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 3 & 1 & 3 & 6 & 3 & 1 \\
3 & 1 & 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 3 & 6 & 3 \\
3 & 3 & 1 & 3 & 3 & 1 & 3 & 3 & 1 & 1 & 3 & 1 & 1 & 3 & 6
\end{array}\right)$$

Define a $b \times b$ matrix $S = [s_{ij}]$ such that

$$s_{ij} = \text{ sum of all the entries in } W_{ij}.$$

Since $B^2 = (k-\mu)I + \mu J + (\lambda - \mu)B$, we have $W_{ij} = \delta_{ij}(k-\mu)I + \mu J + (\lambda-\mu)B_{ij}$. Please note that in the second equation, the dimension of I is $n_i \times n_i$ and the dimension of J is $n_i \times n_j$. We have

$$s_{ij} = \delta_{ij}(k - \mu)n_j + \mu n_i n_j + (\lambda - \mu)c_{ij}n_j. \tag{24}$$

In our example, the matrix is calculated as follows:

$$S = \begin{pmatrix} 70 & 45 & 65 \\ 45 & 90 & 45 \\ 65 & 45 & 70 \end{pmatrix}.$$

The matrix S can also be calculated in a different way.

Lemma 3.1

$$CNR = S. \tag{25}$$

Proof Let α_i be a vector of size v such that

$$\alpha_i(j) = \begin{cases} 1 & \text{if } j \in O_i, \\ 0 & \text{otherwise.} \end{cases}$$

The vectors are chosen such that $s_{ij} = \alpha_i B^2 \alpha_j^T$. Define η_i, $1 \leq i \leq b$, to be a row vector of size v such that $\eta_i(k) = c_{ij}$ for $k \in O_j$. Similarly, define β_j, $1 \leq j \leq b$, to be a row vector of size v such that $\beta_j(k) = r_{ij}$ for $k \in O_i$. We have

$$\begin{aligned} s_{ij} &= \alpha_i B^2 \alpha_j^T = (\alpha_i B)(B\alpha_j^T) \\ &= \eta_i \beta_j^T = \sum_{k=1}^{b} c_{ik} r_{kj} n_k = (CNR)_{ij}. \end{aligned}$$

□

Continuing our example, we calculate s_{12} for the matrix B, using the previous proof, we have

$$\alpha_1 = (11111\ 00000\ 00000),$$

$$\alpha_2 = (00000\ 11111\ 00000),$$

$$\eta_1 = (22222\ 33333\ 11111),$$

and

$$\beta_2 = (33333\ 00000\ 33333).$$

We can see that

$$s_{12} = \alpha_1 B^2 \alpha_2^T = \eta_1 \beta_2^T = 45.$$

3.4 Properties of orbit matrices and prototypes

Lemma 3.1 allows us to derive a set of integer equations for the possible entries in row r of the matrix C. The only information we need are the parameters of the strongly regular graph G; therefore, these equations are independent of the matrix B. Solving these equations will help us to find the orbit matrix C without knowing the matrix B.

Using Equation 25, we have

$$s_{rr} = \sum_{k=1}^{t} c_{rk} r_{kr} n_k = \sum_{k=1}^{t} c_{rk}^2 n_k. \tag{26}$$

Using Equation 24, we have

$$s_{rr} = (k - \mu) n_r + \mu n_r^2 + (\lambda - \mu) c_{rr} n_r. \tag{27}$$

For simplicity, we restrict ourself to the case where the orbits are either of size one or of size p, a prime. Let ψ be the number of orbits of size p and ϕ be the number of orbits of size one. ϕ and ψ satisfy the following equation.

$$\phi = v - p\psi \tag{28}$$

In this case, we have two types of rows (columns). Fixed rows (columns) are those whose orbit size is one and non-fixed rows (columns) are those with orbit size p.

Without taking into account the ordering of the entries in a row of C, we first consider the distribution of such entries. We call this a prototype of a row of C. Each prototype will tell us the possible number of occurrences of each integer as an entry of a particular row of C.

Consider an arbitrary fixed row r of C. The possible value of each entry of that row, regardless of being a fixed column or non-fixed column, is either 0 or 1.

Let x_0 and x_1 be the number of zeros and ones respectively on the fixed columns of row r. Let y_0 and y_1 be the number of zeros and ones respectively on the non-fixed columns of row r. Since the number of fixed columns is ϕ, we have $x_0 + x_1 = \phi$. Similarly we have $y_0 + y_1 = \psi$. Since the row sum of the matrix B is equal to k, we have

$$x_1 + p y_1 = k.$$

Thus, we have the following set of equations:

$$\begin{array}{rcrcrcrl} x_0 & + & x_1 & & & & & = \phi, \\ & & & & y_0 & + & y_1 & = \psi, \\ & & x_1 & & & + & py_1 & = k. \end{array} \tag{29}$$

We define a *Fixed Prototype* as a non–negative integer solution of x_0, x_1, y_0, and y_1, satisfying this set of linear equations.

Now consider an arbitrary non–fixed row r of C. The possible values of the fixed column entries of row r are either 0 or p. The possible values of the non–fixed column entries of row r can be $0, 1, \ldots, p$. Let x_0 and x_p be the number of zeros and p's on the fixed columns of row r. Let y_i, $i = 0, 1, \ldots, p$ be the number of i's on the non–fixed columns of row r. Similar to the situation with columns, we have $x_0 + x_p = \phi$ and $\sum_{i=0}^{p} y_i = \psi$. Also, since the row sum of B is equal to k, by counting, we have $x_p + \sum_{i=1}^{p} iy_i = k$. Using Equation 26, we have

$$p^2 x_p + \sum_{i=1}^{p} i^2 p y_i = s_{rr}.$$

Thus, we have the following set of equations:

$$\begin{array}{rcrrcrcrcrcccrl} x_0 & + & x_p & & & & & & & & & & & & = \phi, \\ & & & y_0 & + & y_1 & + & y_2 & + & y_3 & + & \cdots & + & y_p & = \psi, \\ & & x_p & & + & y_1 & + & 2y_2 & + & 3y_3 & + & \cdots & + & py_p & = k, \\ & & px_p & & + & y_1 & + & 4y_2 & + & 9y_3 & + & \cdots & + & p^2 y_p & = s_{rr}/p. \end{array} \tag{30}$$

We define a *Non–Fixed Prototype* as a non–negative integer solution of x_0, x_p, $y_0, \ldots, y_p$, satisfying this set of linear equations.

In the above equation, the value of s_{rr} is obtained from Equation 27. Using Equation 27, we can generate a separate set of equations for each value of c_{rr}. The following lemma puts a restriction on these values.

Lemma 3.2 *If n_r is odd, then c_{rr} is even.*

Proof Let Y be the subgraph induced by O_i. B_{rr} is the adjacency matrix of Y. Y is a regular graph of degree c_{rr}. By counting the number of edges of Y in two different ways, we have

$$2|E(Y)| = n_r c_{rr}.$$

Since n_r is odd, c_{rr} must be even. $\square$

Since $\phi = v - p\psi$, the smallest possible value for ϕ is $z = v \mod p$. The possible values of ϕ are z, $z+p$, $z+2p$, $\ldots$, $z + p\left\lfloor \frac{v-z}{p} \right\rfloor$. The following two theorems state that, once we find no fixed (or non–fixed) prototype for a given ϕ, then there is no need to consider any larger ϕ's when $\phi \geq 2p$.

Theorem 3.3 *If there exists a fixed prototype with ϕ fixed columns and $\phi \geq 2p$, then there is a fixed prototype with $\phi - p$ fixed rows.*

Proof Since there exists a fixed prototype with ϕ fixed columns, there is an integer solution (x_0, x_1, y_0, y_1) for Equation 29. Consider the following equations for a fixed prototype with $\phi - p$ fixed rows:

$$\begin{array}{ccccccl} x_0' & + & x_1' & & & = & \phi - p, \\ & & & y_0' & + \quad y_1' & = & \psi + 1, \\ & & x_1' & & + \quad py_1' & = & k. \end{array} \tag{31}$$

If $x_0 \geq p$, then $x_0' = x_0 - p$, $x_1' = x_1$, $y_0' = y_0 + 1$, and $y_1' = y_1$ would be a solution for Equation 31, and x_0', x_1', y_0', and y_1' are all non–negative.

If $x_1 \geq p$, then $x_0' = x_0$, $x_1' = x_1 - p$, $y_0' = y_0$, and $y_1' = y_1 + 1$ would be a solution for Equation 31, and x_0', x_1', y_0', and y_1' are all non–negative.

Since $\phi \geq 2p$, one of these two cases has to be true and Equation 31 has a non–negative integer solution. $\square$

Theorem 3.4 *If there exists a non–fixed prototype with ϕ fixed rows and $\phi \geq 2p$, then there is a non–fixed prototype with $\phi - p$ fixed rows.*

Proof Since there exists a non–fixed prototype with ϕ fixed rows, there is a non–negative integer solution $(x_0, x_p, y_0, y_1, \ldots, y_p)$ for the set of Equations 30. Consider the following equations for a non–fixed prototype with $\phi - p$ fixed rows:

$$\begin{array}{rcccccccccccl} x'_0 & + & x'_p & & & & & & & & & & = \phi - p, \\ & & & y'_0 & + & y'_1 & + & y'_2 & + & y'_3 & + \cdots + & y'_p & = \psi + 1, \\ & & x'_p & & + & y'_1 & + & 2y'_2 & + & 3y'_3 & + \cdots + & py'_p & = k, \\ & & px'_p & & + & y'_1 & + & 4y'_2 & + & 9y'_3 & + \cdots + & p^2y'_p & = s_{rr}/p. \end{array} \tag{32}$$

Since $\phi \geq 2p$, either $x_0 \geq p$ or $x_p \geq p$. If $x_0 \geq p$, then $x'_0 = x_0 - p$, $x'_p = x_p$, $y'_0 = y_0 + 1$, and $y'_i = y_i$ for $1 \leq i \leq p$ would be a solution for Equation 32, and $(x'_0, x'_p, y'_0, y'_1, \ldots, y'_p)$ are non–negative.

If $x_p \geq p$, then $x'_0 = x_0$, $x'_p = x_p - p$, and $y'_i = y_i$ for $0 \leq i \leq p-1$, and $y'_p = y_p + 1$ would be a solution for Equation 32, and $(x'_0, x'_p, y'_0, y'_1, \ldots, y'_p)$ are non–negative.

Since $\phi \geq 2p$, one of these two cases has to be true and Equation 32 has a non–negative integer solution.

□

3.5 Upper bounds on the number of fixed points

In this section, we introduce some new upper bounds on ϕ the number of fixed points of an automorphism of a strongly regular graph. We need to use the concept of orbit matrices to derive some of these upper bounds, but some upper bounds are obtained independently, without the use of orbit matrices.

Let B be the adjacency matrix of $G = \text{srg}(v, k, \lambda, \mu)$, having a non-trivial automorphism. Let $B' = [b'_{ij}]$ be the adjacency matrix of the subgraph of G, induced by all the non–fixed vertices. Let $\alpha = \max(\lambda, \mu)$. The following lemma gives a restriction on row–sums of B'.

Lemma 3.5 *Let $G = \text{srg}(v, k, \lambda, \mu)$ and further assume that G has a non-trivial automorphism ρ. Let H be the subgraph of G, induced by all the non-fixed vertices of*

G. Then

$$\delta(H) \geq k - \max(\lambda, \mu).$$

Proof Let x be a non–fixed vertex of G and let $y = x\rho$. Let z be any fixed vertex of G adjacent to x. Since $(z, x) \in E(G)$, we have

$$(z\rho, x\rho) = (z, y) \in E(G).$$

Thus any fixed vertex adjacent to x is adjacent to y as well. Since x and y have at most $\max(\lambda, \mu)$ common neighbours, there are at most $\max(\lambda, \mu)$ fixed vertices adjacent to x. Since G is a regular graph of degree k, every non–fixed vertex of G has at least $k - \max(\lambda, \mu)$, non–fixed neighbours. Thus

$$\delta(H) \geq k - \max(\lambda, \mu).$$

□

Lemma 3.6 *Let $A = [a_{ij}]$ be an $n \times n$ positive semidefinite matrix. Then*

$$\sum_{i,j} a_{ij} \geq 0.$$

Proof Since A is positive semidefinite, we have

$$z^* A z \geq 0,$$

for any $z \in \mathbb{C}^n$. Take the all one vector $\mathbf{j}$ as z, the result follows directly. □

Lemma 3.6 is a special case of the Fejer's theorem. For more information about Fejer's theorem, refer to [28, page 459].

The following theorem gives us an upper bound on the number of fixed points. The interesting fact about this bound is that it is independent from the selection of p.

Theorem 3.7 *Let $r < s < k$ be the eigenvalues of an* $\operatorname{srg}(v, k, \lambda, \mu)$*. then*

$$\phi \leq \frac{\max(\lambda, \mu)}{k - s} v.$$

Proof Let B be the adjacency matrix of the strongly regular graph and $B' = [b'_{ij}]$ be the principal submatrix of B corresponding to the non–fixed vertices.

We know that the idempotent matrix E_1 in Equation 11 is positive semidefinite. Therefore, every principal submatrix of E_1 is positive semidefinite. Let $E'_1 = [e'_{ij}]$ be the submatrix of E_1 corresponding to the non–fixed vertices. Let us further assume that E'_1 is of size n. Let $\alpha = \max(\lambda, \mu)$. From Lemma 3.5, we know that the row sum of the matrix B' is at least $k - \alpha$. Therefore,

$$\sum_{ij} b'_{ij} \geq (k - \alpha)n. \tag{33}$$

In order for E'_1 to be positive semidefinite, using Lemma 3.6, we have

$$\sum_{ij} e'_{ij} \geq 0. \tag{34}$$

By considering that $r - s < 0$ and combining Equations 11, 33, and 34, we have

$$\begin{aligned} \frac{1}{r-s}\left\{(k-\alpha)n - sn + \frac{s-k}{v}n^2\right\} &\geq \frac{1}{r-s}\left\{\sum_{ij} b'_{ij} - sn + \frac{s-k}{v}n^2\right\} \\ &= \sum_{ij} e'_{ij} \\ &\geq 0. \end{aligned}$$

Since $r - s < 0$ and $n > 0$, we have

$$(k - \alpha) - s + \frac{s-k}{v}n \leq 0.$$

After simplifying, we get

$$n \geq v - \frac{\alpha v}{k - s},$$

which implies

$$\phi \leq \frac{\alpha v}{k - s}.$$

□

As an example, consider an srg(99, 14, 1, 2), using Theorem 3.7, we can see that, an automorphism of this strongly regular graph, can have at most 18 fixed points.

Theorem 3.8
$$\phi \leq v - \frac{k^2 - k}{\max(\lambda, \mu)} + 2k - \max(\lambda, \mu) - 2.$$

Proof Let $\alpha = \max(\lambda, \mu)$. Let B be the adjacency matrix of the strongly regular graph and $B' = [b'_{ij}]$ be the principal submatrix of B corresponding to the non–fixed vertices. Let n be the size of B'. Using Lemma 3.5, the matrix B' has at least $k - \alpha$ ones in each column. Since any two vertices have at most α common neighbours, we have:

$$B'^2 \leq (k - \alpha)I + \alpha J. \tag{35}$$

We count the sum of entries of B'^2 in two different ways. First, since B' is symmetric, we have $B'^2 = B'B'^T$. Let r be an arbitrary column of B'. By Lemma 3.5 there are at least $k - \alpha$ ones on r. By counting the number of un–ordered pairs of 1's on the column r, we have

$$\sum_{1 \leq i,j \leq n, i \neq j} b'_{ir} b'_{jr} \geq (k - \alpha)(k - \alpha - 1).$$

Let s be the sum of all the entries of B'^2, except the ones on the diagonal. In fact, s is the sum of inner products of different rows. We have

$$\begin{aligned} s &= \sum_{1 \leq i,j \leq n, i \neq j} \sum_{r=1}^{n} b'_{ir} b'_{jr} \\ &= \sum_{r=1}^{n} \sum_{1 \leq i,j \leq n, i \neq j} b'_{ir} b'_{jr} \\ &\geq n(k - \alpha)(k - \alpha - 1). \end{aligned}$$

Using Equation 35, we have

$$s \leq \alpha n(n - 1).$$

Using the last two calculations, we have

$$(k - \alpha)(k - \alpha - 1)n \leq \alpha n(n - 1).$$

One could simplify the above equation to see

$$n \geq \frac{k^2 - k}{\alpha} - 2k + \alpha + 2.$$

Since $n = v - \phi$, we have

$$\phi \leq v - \frac{k^2 - k}{\alpha} + 2k - \alpha - 2.$$

□

As an example, again consider an srg(99, 14, 1, 2), using Theorem 3.8, we can see that, an automorphism of this strongly regular graph, can have at most 32 fixed points.

We have realised, in all our test cases, Theorem 3.7 gives a better upper bound than Theorem 3.8, but we have not been able to prove that the upper bound obtained from Theorem 3.7 is always lower than the upper bound obtained from Thorem 3.8.

The following theorem states that when the orbit size is large enough, there is no fixed point.

Theorem 3.9 *If $p > k$ and $\mu \neq 0$, then $\phi = 0$.*

Proof Consider Equation 29. Since $p > k$, the only solution to the equation is the following:

$$(x_0, x_1, y_0, y_1) = (\phi - k, k, \psi, 0).$$

Let u be a fixed vertex and v be a non–fixed vertex. Since $y_1 = 0$, we have $B_{uv} = 0$. Consider an arbitrary vertex x different from u and v. Again, since $y_1 = 0$, if x is a fixed vertex, we have $B_{xv} = 0$. If x is a non–fixed vertex, we have $B_{xu} = 0$. In both cases x is not a common neighbour of u and v. Thus either $\mu = 0$ or there is no fixed point. □

Corollary 3.10 *If $p > k$ and $\mu \neq 0$, then v should be divisible by p.*

The following theorems put some bound on the number of fixed points a strongly regular graph can have.

Theorem 3.11 *If $\phi \leq k - 1$ and $p > \max(\lambda, \mu)$, then $\phi < v/(p + 1)$.*

Proof Using Equation 29, since $x_1 \leq \phi < k$, we have $py_1 \geq 1$. Therefore, $y_1 \geq 1$. Consider the submatrix of C corresponding to the fixed rows and non–fixed columns and call it C'. Since $y_1 \geq 1$, there is at least one "1" in each row of C'. Since C' is of size $\phi \times \psi$ there are at least ϕ ones in C'. If $\phi > \psi$, by the pigeon hole principle, there is a column of C' that has more than one 1 in it. Therefore there are two fixed rows, i and j, and a non–fixed column k such that $c_{ik}c_{jk} = 1$. But, we have $pc_{ik}c_{jk} \leq \max(\lambda, \mu)$ since the inner product of two fixed rows of B is less than $\max(\lambda, \mu)$, which contradicts the assumption that $p > \max(\lambda, \mu)$. Therefore $\phi \leq \psi$. Since $\phi + p\psi = v$, we have $\phi < v/(p+1)$. □

3.6 Computer construction of orbit matrices

After we have obtained the prototypes, we can construct the matrix C by backtracking. The first row can be constructed easily using its prototype. For the first row we know the number of occurrences of each value. The order of entries is not important for the first row as the orbits of the same type are still free to be permuted among themselves.

The rest of the rows can be constructed recursively. Assume that the matrix C is constructed up to the row $r-1$. To construct row r, first we consider the prototype. Using Equation 24, for $1 \leq i < r$, we have

$$s_{ir} = \mu n_i n_r + (\lambda - \mu) c_{ir} n_r. \tag{36}$$

Since $i < r$, the value of c_{ir} is already known. Hence the value of s_{ir} is known. On the other hand, using Equation 25, we have

$$s_{ir} = \sum_{k=1}^{b} c_{ik} c_{rk} n_r. \tag{37}$$

The new row r has to satisfy Equation 37 for all $1 \leq i < r$. The entries in row r are constructed recursively starting from column 1 and ending at column b. Since the variables on the right–hand side of Equation 37 are all non–negative, if we are at a column $b' < b$, Equation 37 can be replaced by

$$s_{ir} \geq \sum_{k=1}^{b'} c_{ik} c_{rk} n_r. \tag{38}$$

For each column b', we try all possible values for $c_{rb'}$, but ensuring that it satisfies Equation 38.

Since B is an adjacency matrix, it is symmetrical. Thus, for $k < r$, $c_{rk} = c_{kr}(n_k/n_r)$, which is already known.

The number of candidate matrices for C can further be reduced by isomorph rejection, since the fixed and non–fixed columns can permute among themselves respectively.

We finish this chapter using an example, to see how the orbit matrices for strongly regular graphs are constructed.

Example 9. In this example, we consider srg$(15, 6, 1, 3)$. Assume $p = 3$ and assume that there are three orbits of size one and four orbits of size three. Therefore $\phi = 3$, $\psi = 4$, $n_1 = n_2 = n_3 = 1$ and $n_4 = n_5 = n_6 = n_7 = 3$.

First, we calculate the fixed prototype using Equation 29. We have the following equations:

$$\begin{aligned} x_0 + x_1 \quad\quad\quad\quad &= 3, \\ y_0 + y_1 &= 4, \\ x_1 \quad\quad + 3y_1 &= 6. \end{aligned} \tag{39}$$

The only non–negative integer solutions of the above equations are

$$(x_0, x_1, y_0, y_1) \in \{(0, 3, 3, 1), (3, 0, 2, 2)\}.$$

Since the diagonal of B is zero, $c_{rr} = 0$ when r is a fixed row. Therefore, there has to be at least one zero in the fixed columns. Thus $x_0 \geq 1$ and the first solution is not acceptable, implying $(x_0, x_1, y_0, y_1) = (3, 0, 2, 2)$. Now, we consider the non–fixed prototype for the non–fixed rows of the incidence matrix. Equation 30 gives:

$$\begin{array}{ccccccccccc} x_0 & + & x_3 & & & & & & & = 3, \\ & & & & y_0 & + & y_1 & + & y_2 & + & y_3 & = 4, \\ & & x_3 & & & + & y_1 & + & 2y_2 & + & 3y_3 & = 6, \\ & & 3x_3 & & & + & y_1 & + & 4y_2 & + & 9y_3 & = s_{rr}/3. \end{array} \tag{40}$$

Using Equation 27, we have $s_{rr} = 36 - 6c_{rr}$. The possible values of c_{rr} are 0, 1, and 2. We shall consider all these cases.

For $c_{rr} = 0$, we have $s_{rr} = 36$. The solutions of Equation 40 are

$$\begin{aligned} (x_0, x_3, y_0, y_1, y_2, y_3) \in \{ & (0, 3, 1, 3, 0, 0), \\ & (1, 2, 1, 2, 1, 0), \\ & (2, 1, 1, 1, 2, 0), \\ & (3, 0, 1, 0, 3, 0), \\ & (3, 0, 0, 3, 0, 1) \ \}. \end{aligned}$$

Since $c_{rr} = 0$, we have $y_0 > 0$. Therefore the last solution is not possible.

We do not need to consider the case $c_{rr} = 1$ using Lemma 3.2.

For $c_{rr} = 2$, we have $s_{rr} = 24$. In this case the prototype in 40 has no non–negative integer solution.

Now, we build the matrix C row by row. The first row is a fixed row. Its only prototype is $(3, 0, 2, 2)$, meaning 3 zeros for the fixed columns, and 2 zeros and 2 ones among the non–fixed columns. By permuting the columns, we can assume that the first row is $[000|0011]$.

Now, we need to construct the second row. Since $c_{12} = 0$, Equation 36 implies

$$s_{12} = 3 - 2c_{12} = 3.$$

Thus by Equation 37, we have

$$\sum_{k=1}^{7} c_{1k} c_{2k} n_k = 3.$$

Since we already know the first row of C, we have:

$$3c_{26} + 3c_{27} = 3. \tag{41}$$

The prototype for the second row is $(3, 0, 2, 2)$. Considering Equation 41 and, by permuting the columns, we can assume that the second row is $[000|0101]$. By a similar argument the third row is $[000|1001]$ up to isomorphism.

Consider the forth row. By Equation 36, we have

$$\begin{aligned} s_{14} &= 9 - 6c_{14} = 9, \\ s_{24} &= 9 - 6c_{24} = 9, \\ s_{34} &= 9 - 6c_{34} = 3. \end{aligned}$$

Considering the values of s_{i4}, by Equation 37, we have:

$$\begin{aligned} 3c_{46} + 3c_{47} &= 9, \\ 3c_{45} + 3c_{47} &= 9, \\ 3c_{44} + 3c_{47} &= 3. \end{aligned}$$

The only solutions of the above set of equations are

$$(c_{44}, c_{45}, c_{46}, c_{47}) \in \{ \; (1, 3, 3, 0), \; (0, 2, 2, 1) \; \}.$$

The first solution does not belong to any non–fixed prototype, however the second solution does. Therefore the forth row up to isomorphism is $[003|0221]$.

Similarly, we construct the rest of the rows of C by using their prototypes, Equations 36, 37, and by permuting the rows in order to consider the symmetry.

Finally, up to isomorphism one can see that there is only one possible solution for the column sum matrix, which is the following:

$$C = \left(\begin{array}{ccc|cccc} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 3 & 0 & 2 & 2 & 1 \\ 0 & 3 & 0 & 2 & 0 & 2 & 1 \\ 3 & 0 & 0 & 2 & 2 & 0 & 1 \\ 3 & 3 & 3 & 1 & 1 & 1 & 0 \end{array}\right)$$

$\square$

Chapter 4

Computer search for strongly regular graphs

We have implemented a computer program to search for unknown strongly regular graphs. Our algorithm is an exhaustive backtrack search based on orbit matrices. Throughout the thesis we call this program the SRG program.

After an orbit matrix for the desired strongly regular graph is obtained, the SRG program tries to find the adjacency matrix of the strongly regular graph by expanding the entries of the orbit matrix into blocks of the adjacency matrix.

Several combinatorial and algebraic techniques have been used in pruning the backtrack search tree.

The SRG program also has the ability to estimate the size of the search tree using a random probing method. This random probing technique can be used as well to perform a randomised search.

4.1 History of computer search for strongly regular graphs

The existence or non–existence of strongly regular graphs has been studied using computer searches. It was shown in [5] by the use of a computer search that srg(49, 16, 3, 6) does not exist. The uniqueness of some strongly regular graphs up to isomorphism was shown by the use of an exhaustive search in [15]. Complete classification of some strongly regular graphs with small parameters has been performed in [13], [26], [38], and [44], by the use of computer searches. Corneil and Mathon in [14] describe several algorithmic techniques for the construction of strongly regular graphs and other combinatorial configurations. In this paper general search techniques for combinatorial configurations such as hill–climbing and backtracking, as well as specific techniques for strongly regular graphs such as switching classes are introduced. In [36], a computer search was performed to find all self–complementary strongly regular graphs with less than 54 vertices.

4.2 Methodology

The SRG program tries to complete the adjacency matrix of a strongly regular graph by expanding the entries of its orbit matrix into circulant submatrices. Let us assume that $C = [c_{ij}]$ is the orbit matrix of the strongly regular graph corresponding to an automorphism of order p, and that $B = [B_{ij}]$ is the adjacency matrix of the strongly regular graph. The SRG program would expand each non–fixed upper triangular entry c_{ij} to all the possible $\binom{p}{c_{ij}}$ circulant matrices B_{ij}.

After each circulant block B_{ij} is placed into the matrix B, a check would be applied to see that B does not violate the properties of strongly regular graphs. This is the pruning part of the algorithm. We will discuss these checks in more detail later. Because the adjacency matrix B is symmetric, the lower triangular half of the matrix is obtained by symmetry.

In order to make the backtrack search flexible for each non–fixed upper diagonal entry of C, we associate a time–stamp in the SRG program. The time stamp represents the order which the entries c_{ij} are expanded into circulant blocks B_{ij} in the backtrack search.

The program also has the ability to perform isomorph rejection at a specific given time stamp. It means the user is able to specify at which level of the search tree the isomorph rejection algorithm should be applied. We usually use isomorph rejection at the beginning and at the very end of the search. Isomorph rejection is time consuming and very costly when applied to the middle level where there are many cases.

Another pruning technique that is implemented into the SRG program is the positive semidefinite test. Whenever a principal submatrix is completed, the SRG program checks whether or not the corresponding submatrix is positive semidefinite. As we have shown in Chapter 2, the idempotent matrices E_1 and E_2, as defined in Equations 11 and 12, are positive semidefinite. Therefore their principal submatrices are positive semidefinite as well. This is a strong pruning technique, and has given us the ability to solve cases which we were not able to complete before using this pruning technique. The program uses a maximal clique algorithm to find at which time stamps a new principal submatrix is completed. Let n be the number of orbits and let T_{ij} represent the time stamp of the i, j block. Define the graph $G(V, E)$ as follows:

- $V = \{1 \ldots n\}$;
- $uv \in E$ iff $T_{uu} \leq t, T_{vv} \leq t,$ and $T_{uv} \leq t$.

Each clique in G corresponds to a complete principal submatrix at an arbitrary time stamp t.

In order to find all strongly regular graphs with parameters (v, k, λ, μ) that have an automorphism group of size divisible by a prime p, one should do the following procedure:

1. Find all the orbit matrices with p as the non–fixed orbit size for all the possible fixed points.

2. Run the SRG program for all orbit matrices obtained.

3. Run a final isomorphism test.

4.3 An example

In this section, we show by an example, how the SRG program works and the structure of its input file. We consider the srg$(15, 6, 1, 3)$ for this example, with $p = 5$, and no fixed points. After running the orbit matrix program, we find that there are exactly two orbit matrices as follows:

$$\begin{pmatrix} 0 & 3 & 3 \\ 3 & 0 & 3 \\ 3 & 3 & 0 \end{pmatrix},$$

and

$$\begin{pmatrix} 0 & 3 & 3 \\ 3 & 2 & 1 \\ 3 & 1 & 2 \end{pmatrix}.$$

For this example, we consider the second matrix.

The description of the input file (Figure 1) is as follows:

Line 1: The parameters v, k, λ, μ of the strongly regular graph;

Line 2: Prime p for the size of the non–fixed orbits, and the number of orbits;

Line 3: Orbit sizes (here we have three orbits of size 5);

Lines 4–6: The orbit matrix;

Lines 7–9: The time stamps. Please note that since $T_{ij} = T_{ji}$, only the upper triangular part of the matrix is required;

Line 10: The inverse probability of going into each backtrack level (this can be used either for a random search or estimation);

Line 11: The stop time and snap shot frequency;

```
1:    15 6 1 3
2:    5 3
3:    5 5 5

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

7:    1 4 6
8:      2 5
9:        3

10:   1 1 1 1 1 1
11:   0 0
12:   1
13:   0 1 0 0 0 1
14:   0
```

Figure 1: The input file for this example

Line 12: A boolean control variable to show that we are using manual isomorph rejection;
Line 13: An array which indicates at which time stamp we have isomorph rejection;
Line 14: A boolean control variable to show whether or not, we have a partial entry (in this case we do not).

We ran the program to get a snap shot at $t = 2$. At time stamp $t = 2$, two partial solutions B_1 and B_2 were found.

$$B_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & 0 & 1 & 0 & 0 & 1 & X & X & X & X & X \\ X & X & X & X & X & 1 & 0 & 1 & 0 & 0 & X & X & X & X & X \\ X & X & X & X & X & 0 & 1 & 0 & 1 & 0 & X & X & X & X & X \\ X & X & X & X & X & 0 & 0 & 1 & 0 & 1 & X & X & X & X & X \\ X & X & X & X & X & 1 & 0 & 0 & 1 & 0 & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \end{pmatrix}.$$

$$B_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & 0 & 0 & 1 & 1 & 0 & X & X & X & X & X \\ X & X & X & X & X & 0 & 0 & 0 & 1 & 1 & X & X & X & X & X \\ X & X & X & X & X & 1 & 0 & 0 & 0 & 1 & X & X & X & X & X \\ X & X & X & X & X & 1 & 1 & 0 & 0 & 0 & X & X & X & X & X \\ X & X & X & X & X & 0 & 1 & 1 & 1 & 1 & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X & X & X & X & X & X \end{pmatrix}.$$

Since B_1 is isomorphic to B_2, and we have isomorph rejection at the time stamp $t = 2$, the second solution B_2 would be rejected.

We finished the run to see how many strongly regular graphs were found. This is a excerpt of the output of the program for this example:

```
aut size of the solution: autogp size =(1)(4)(5)
```

```
solution number 1 :
00000 11010 11100
00000 01101 01110
00000 10110 00111
00000 01011 10011
00000 10101 11001

10101 01001 00010
11010 10100 00001
01101 01010 10000
10110 00101 01000
01011 10010 00100

10011 00100 00110
11001 00010 00011
11100 00001 10001
01110 10000 11000
00111 01000 01100


The number of strongly regular graphs found:1
elapsed time = 1.000000 ms
Number of nodes visited= 64
```

1:	1	-	0
2:	1	1	0
3:	10	-	-
4:	25	-	0
5:	25	-	-
6:	1	24	0

```
The total estimation of number of nodes= 64
elapsed time=0.00 seconds
 elapsed time==0.00 hours
 Estimated time = 0.00E+00 days
```

The last part of the output shows the size of the backtrack search tree. From the output, we can see that at time stamp $t = 1$ there is one partial solution, at time stamp $t = 2$ there are two solutions, but one is rejected by the isomorphism test. At time stamp $t = 3$ there are ten partial solutions and the isomorphism test is not performed. At time stamp $t = 6$ there are 25 solutions, but only 1 non-isomorphic solution.

4.4 Correctness tests

In order to make sure that the SRG program works correctly, we performed some tests. The first test was running the program on a few small cases that could be verified by hand. We also compared our results with the results from other people. Since the chance of getting exactly the same result from different algorithms and different implementations is very small, we can conclude, with a high level of confidence, that the SRG program is correct.

One of the test cases that we used was the srg$(36, 14, 4, 6)$. The complete enumeration of this case has been done by McKay and Spence in [38]. There are 180 strongly regular graphs with the above parameters which can be downloaded from

``http://www.maths.gla.ac.uk/~es/srgraphs.html''.

We ran the SRG program for all the possible orbit sizes to compare our results to the results of McKay and Spence. The SRG program found 152 srg$(36, 14, 4, 6)$ with non-trivial automorphism groups.

Table 4 shows the statistics on the number of strongly regular graphs found by Brendan McKay and Edward Spence, and the strongly regular graphs found by the

Automorphism group size	Number of SRGs McKay program	Number of SRGs the SRG program
1	28	Not Applicable
2	37	37
3	14	14
4	51	51
8	16	16
12	5	5
16	5	5
21	2	2
24	9	9
32	1	1
36	1	1
48	5	5
64	1	1
72	1	1
144	1	1
216	1	1
432	1	1
12096	1	1

Table 4: Automorphism group statistics of all srg$(36, 14, 4, 6)$

SRG program. The results are exactly the same except for the asymmetric graphs, which the SRG program is not designed to find. Since it is very unlikely to get the same results randomly, this test shows us that the SRG program works correctly.

Another test that was done can be found in Section 4.6.2.

4.5 Estimations

Since the running time of most of the combinatorial search algorithms, as well as the SRG program, are not polynomial, it is very important to have an estimation of the running time of the program before starting the exhaustive search.

Donald Knuth in [29], showed, for the first time, a simple method for estimating the size of the search tree of a backtrack algorithm. Let us assume that the level i in a search tree is the set of all nodes with distance i to the root. In Knuth's method, a random path $P = P_0 P_1 \dots P_n$ from the root of the search tree to one of the leaves is taken. Let c_i be the number of children of P_i in the search tree. Then an estimate of the number of nodes at level i is calculated as follow:

$$E_i = c_0 c_1 \dots c_{i-1}. \tag{42}$$

An estimate of the size of the search tree is calculated as follows:

$$E = 1 + c_0 + c_0 c_1 + c_0 c_1 c_2 + \cdots + c_0 c_1 c_2 \cdots c_{n-1}. \tag{43}$$

Under the assumption that all children of a node have equal probability of being chosen, Donald Knuth in [29] proved that the expected value of E in Equation 43 is equal to the size of the search tree. Therefore if we compute E for various times and calculate the average, we can get a good estimation of the size of the backtrack tree. For more information about backtracking, backtrack search tree, and the Knuth's method, refer to [30].

In the SRG program, we applied a method, similar to the Knuth's method, to estimate the size of the search tree. This method has some advantages to the Knuth's method which will be explained later.

We assign the probabilities p_i to each level of the backtrack tree. In the backtrack search, if we are at a node X at level i, we visit each child of X with probability p_i. If all p_i's are equal to 1, we visit all the nodes of the backtrack search tree, thus we do a complete exhaustive search. Therefore by the choice of p_i, we can do either a complete search, or a random search using the same program. If we do a random search, we calculate the estimated number of nodes at each level of the backtrack search using a recursive method. Let X be a node of the backtrack search tree at level t that has been visited in our random search. Let $\mathcal{C}(X)$ be all the children of X that have been visited in the random search. Define:

$$\mathcal{E}_i(X) = \begin{cases} 1 & \text{if } i = t, \\ 0 & \text{if } i < t, \\ \sum_{Y \in \mathcal{C}(X)} \mathcal{E}_i(Y) p_i & \text{otherwise.} \end{cases} \tag{44}$$

Define

$$E_i = \mathcal{E}_i(\text{ROOT}),$$

then E_i is the estimated number of nodes at each level i, and $\sum_i E_i$ is the estimated number of all the nodes of the backtrack search.

This method fits perfectly into the backtrack program since it is recursive. We calculate $\mathcal{E}_i(X)$ whenever we visit a node X in the backtrack search. The proof of this method is similar to the proof of the Knuth's method. For the proof of the Knuth's method refer to [30, page 117].

If we do the complete search, then E_i would be the exact number of nodes at level i. In this method, we have control over the choice of the probabilities p_i. We can visit the nodes at some levels, especially the levels near the root and near the leaves, more often than other levels of the backtrack search. This would gives us a better estimation of the size of the backtrack search.

4.6 Results

In this section, we provide all the results of running the genOrbit and the SRG program. We divided the results into two subsections. One section is related to the strongly regular graphs whose existence or non-existence is unknown. The other section provides the results about the strongly regular graphs whose existence is known, but there has not been a complete classification performed.

4.6.1 Results on unknown strongly regular graphs

In this section, we are investigating the strongly regular graphs whose existence is unknown. We obtained the list of these strongly regular graphs from the CRC handbook of combinatorial designs [11].

The details of all the computer runs on these graphs can be found in Appendix A.

Theorem 4.1 *If an* srg$(65, 32, 15, 16)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, *and* 5. *Moreover, if it has an automorphism of order* 5, *then it can only have* 5 *fixed points.*

Theorem 4.2 *If an* srg$(69, 20, 7, 5)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2 *and* 3.

Theorem 4.3 *If an* srg$(75, 32, 10, 16)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2 *and* 3.

Theorem 4.4 *If an* srg$(76, 30, 8, 14)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2 *and* 3.

Theorem 4.5 *If an* srg$(76, 35, 18, 14)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, *and* 5. *Moreover, if it has an automorphism of order* 5, *then it can only have* 1 *fixed point.*

Theorem 4.6 *If an* srg$(85, 14, 3, 2)$ *exists, the only possible prime devisor of the size of its automorphism group is* 2.

Theorem 4.7 *If an* srg$(85, 30, 11, 10)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, 5, *and* 17.

Theorem 4.8 *If an* srg$(85, 42, 20, 21)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, 5, *and* 7.

Theorem 4.9 *If an* srg$(88, 27, 6, 9)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, 5, *and* 11.

Theorem 4.10 *If an* srg$(95, 40, 12, 20)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, *and* 5.

Theorem 4.11 *If an* srg$(96, 35, 10, 14)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, *and* 5. *Furthermore, if it has an automorphism of order* 5, *then it has no more than one fixed point.*

Theorem 4.12 *If an* srg$(96, 38, 10, 18)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, *and* 5.

Theorem 4.13 *If an* srg$(96, 45, 24, 18)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, *and* 5.

Theorem 4.14 *If an* srg$(99, 14, 1, 2)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2 *and* 3. *Moreover, if it has an automorphism of order* 3, *then it has no fixed points.*

Theorem 4.15 *If an* srg$(99, 42, 21, 15)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, 5, 7, *and* 11.

Theorem 4.16 *If an* srg$(100, 33, 8, 12)$ *exists, the only possible prime devisors of the size of its automorphism group are* 2, 3, 5, *and* 11.

Table 5 summarises the results.

4.6.2 Results on known strongly regular graphs

We have used the SRG program on parameter sets where there is a strongly regular graph known, for three reason:

1. To test the SRG program;
2. To find new strongly regular graph that are not isomorphic to any of the known ones;
3. To build a database of strongly regular graphs with non–trivial automorphisms.

One of the strongly regular graphs that we have studied is srg$(49, 18, 7, 6)$. Using our computer program, we have generated all srg$(49, 18, 7, 6)$ which have automorphisms of order divisible by 5 or 7.

It is mentioned in [11] that all known srg$(49, 18, 7, 6)$ are either from OA$(7, 3)$ or Pasechnik(7).

G	possible primes $\{p : p \vert Aut(G)\}$
srg(65, 32, 15, 16)	2,3,5
srg(69, 20, 7, 5)	2,3
srg(75, 32, 10, 16)	2,3
srg(76, 30, 8, 14)	2,3
srg(76, 35, 18, 14)	2,3,5
srg(85, 14, 3, 2)	2
srg(85, 30, 11, 10)	2,3,5,17
srg(85, 42, 20, 21)	2,3,5,7
srg(88, 27, 6, 9)	2,3,5,11
srg(95, 40, 12, 20)	2,3,5
srg(96, 35, 10, 14)	2,3,5
srg(96, 38, 10, 18)	2,3,5
srg(96, 45, 24, 18)	2,3,5
srg(99, 14, 1, 2)	2,3
srg(99, 42, 21, 15)	2,3,5,7,11
srg(100, 33, 8, 12)	2,3,5,11

Table 5: Results summery on the automorphism groups of unknown strongly regular graphs

We have reviewed both of the above constructions in Chapter 2. OA(7, 3) is equivalent to a Latin square of order 7. According to [11] there are exactly 147 Latin squares of order 7. We have obtained all the non–isomorphic Latin squares of order 7 from Professor Brendan McKay's webpage at

`` http://cs.anu.edu.au/~bdm/data/latin.html ''.

We generated all the strongly regular graphs obtained from these 147 Latin squares, and compared them to our own results. Table 6 shows the automorphism group size statistics of all the strongly regular graphs obtained from Latin squares of order 7.

It is mentioned in [11] that all known srg(49, 18, 7, 6) are either obtained from Latin squares or from the Pasechnik method. We wrote a computer program to find all Pasechnik srg(49, 18, 7, 6).

It can be seen by hand calculations that there is exactly one skew symmetric Hadamard matrix of order 8 up to isomorphism which is the following:

Automorphism group size	Number of strongly regular graphs
1	44
2	57
3	4
4	11
6	16
8	1
10	1
12	2
15	1
16	2
18	1
24	3
72	1
144	1
1008	1
1764	1

Table 6: Automorphism group statistics of all srg$(49, 18, 7, 6)$ obtained from Latin squares of order 7

Automorphism group size	Number of strongly regular graphs
10	1
15	3
21	1
30	1
63	1
126	1
1008	1
1764	1

Table 7: Automorphism group size statistics of all srg$(49, 18, 7, 6)$ with automorphism group size divisible by 5 and 7 obtained from the SRG program.

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ - & 1 & 1 & 1 & 1 & - & - & - \\ - & - & 1 & 1 & - & - & 1 & 1 \\ - & - & - & 1 & 1 & 1 & 1 & - \\ - & - & 1 & - & 1 & 1 & - & 1 \\ - & 1 & 1 & - & - & 1 & 1 & - \\ - & 1 & - & - & 1 & - & 1 & 1 \\ - & 1 & - & 1 & - & 1 & - & 1 \end{pmatrix}$$

There are two srg$(49, 18, 7, 6)$ that can be obtained from the matrix H above by the Pasechnik method. We compared these two graphs to the Latin square srg$(49, 18, 7, 6)$. We realised that the two Pasechnik graphs are isomorphic to a strongly regular graph with automorphism group of size 1764 which can be obtained from a Latin square.

Table 7 shows all the strongly regular graphs with automorphism group of size divisible by 5 and 7.

We compared the output of our program for $p = 5$ and $p = 7$ to all srg$(49, 18, 7, 6)$ from Latin squares of order 7 and obtained the following two important conclusions:

- All Latin square srg$(49, 18, 7, 6)$ which had automorphism orders divisible by 5 and 7 were found by the SRG program, which is a strong correctness test of our program.

- We have found 6 new strongly regular graphs that were not known before. The graphs can be found in Appendix B.

Chapter 5

Partial geometries

5.1 Introduction

Theorem 2.29 on page 41 shows that the point graph of a partial geometry is strongly regular. Since the line graph of a partial geometry is the point graph of its dual, the line graph is also strongly regular. Table 2 on page 8 shows some unknown partial geometries, and the parameters of their associated point and line graphs. It is tempting to construct partial geometries from their associated strongly regular graphs. In this chapter, we develop the theory of orbit matrices for partial geometries, and give some preliminary results.

5.2 Automorphisms of partial geometries

Let A be the incidence matrix of a partial geometry $\mathrm{pg}(s, t, \alpha)$ and let B the adjacency matrix of the associated point graph. Equation 23 gives

$$B = AA^T - (t+1)A.$$

An automorphism of A is a pair of permutation matrices P and Q such that

$$PAQ = A.$$

Here P permutes the rows (points) and Q the columns (lines). We have

$$\begin{aligned} PBP^T &= P(AA^T - (t+1)I)P^T \\ &= PAA^TP^T - (t+1)I \\ &= PAQQ^TA^TP^T - (t+1)I \\ &= AA^T - (t+1)I \\ &= B. \end{aligned}$$

Therefore, P is the automorphism of the point graph. Similarly Q is the automorphism of the line graph. One should note that an automorphism of the point or line graph need not be an automorphism of the partial geometry. For example, when $\alpha = s+1$, the partial geometry is a $2-(v, s+1, 1)$ design. The point graph is a complete graph and any permutation is an automorphism of the complete graph, but not necessarily an automorphism of the design.

We next see how the assumption of the existence of a non-trivial automorphism can help to reduce the size of the search for possible point and line graphs of a partial geometry. Later on, we shall also see how the automorphism can be used to find a partial geometry, given a line graph.

5.3 Orbit matrices for partial geometries

Once we have an orbit matrix C for B, we can find the possible orbit matrices of A. Assume the columns of A are in orbits of size $m_1, m_2, \ldots, m_d$. For simplicity, we restrict ourself again to the case that we only have column orbits of size 1 or a prime p.

Let v' be the number of columns of A. Let η_p be the number of column–orbits of size p and $\eta_1 = v' - p\eta_p$ be the number of column–orbits of size 1.

Let $C_A = [u_{ij}]$ be the $b \times d$ column-sum orbit matrix of A and $C_{AA^T} = [v_{ij}]$ be the $b \times b$ column–sum orbit matrix of AA^T. Using Equation 23, we have

$$C_{AA^T} = C + (t+1)I. \tag{45}$$

Since the row sum is $t+1$, we have

$$\sum_{j=1}^{d} u_{ij}\left(\frac{m_j}{n_i}\right) = t+1. \tag{46}$$

Checking the column sum, we get

$$\sum_{i=1}^{b} u_{ij} = s+1. \tag{47}$$

Let $M = \mathrm{diag}(m_1, m_2, \ldots, m_d)$. Using a method similar to the one in Lemma 3.1, one can show that

$$C_A M C_A^T = S_{AA^T}, \tag{48}$$

where the (i,j)–th entry of S_{AA^T} is the sum of all the entries in the (i,j)–th block of the matrix AA^T, which is in fact equal to $v_{ij}n_j$. Since M is a diagonal matrix, using Equation 48, we have

$$\sum_{k=1}^{d} u_{ik}u_{jk}m_k = v_{ij}n_j. \tag{49}$$

More specifically, when $i=j$, Equation 49 implies

$$\sum_{k=1}^{d} u_{ik}^2 m_k = v_{ii}n_i. \tag{50}$$

Using Equations 46 and 50, we define the fixed and non–fixed prototypes for the matrix C_A, in a way similar to the prototypes for the matrix C.

Consider an arbitrary fixed row r of C_A. Let w_0 and w_1 be the number of zeros and ones, respectively, on the fixed columns of row r. Let z_0 and z_1 be the number of zeros and ones, respectively, on the non–fixed columns of row r. Since the number of fixed columns is η_1, we have $w_0 + w_1 = \eta_1$. Similarly we have $z_0 + z_1 = \eta_p$. Since the row sum of the matrix A is equal to $t+1$, we have

$$w_1 + pz_1 = t+1.$$

Thus, we have the following set of equations:

$$\begin{array}{ccccccl} w_0 & + & w_1 & & & & = \eta_1, \\ & & & z_0 & + & z_1 & = \eta_p, \\ & & w_1 & & + & pz_1 & = t+1. \end{array} \tag{51}$$

We define a *Fixed Prototype* as a non–negative integer solution to these linear equations.

Now consider an arbitrary non–fixed row r of C_A. The possible values of the fixed column entries of row r are either 0 or p. The possible values of the non–fixed column entries of row r can be $0, 1, \ldots, p$. Let w_0 and w_p be the number of zeros and p's on the fixed columns of row r. Let z_i, $i = 0, 1, \ldots, p$, be the number of i's on the non–fixed columns of row r. Similar to the situation with fixed rows, we have $w_0 + w_p = \eta_1$ and $\sum_{i=0}^{p} z_i = \eta_p$. Also, since the row sum of A is equal to $t+1$, by counting we have $w_p + \sum_{i=1}^{p} iz_i = t+1$. Using Equation 50, we have

$$p^2 w_p + \sum_{i=1}^{p} i^2 p z_i = v_{rr} n_r = p v_{rr} = p(c_{rr} + t + 1).$$

Thus, we have the following set of equations:

$$\begin{array}{ccccccccccccccl} w_0 & + & w_p & & & & & & & & & & & & = \eta_1, \\ & & & & z_0 & + & z_1 & + & z_2 & + & z_3 & + \cdots + & z_p & & = \eta_p, \\ & & w_p & & & + & z_1 & + & 2z_2 & + & 3z_3 & + \cdots + & pz_p & & = t+1, \\ & & pw_p & & & + & z_1 & + & 4z_2 & + & 9z_3 & + \cdots + & p^2 z_p & & = c_{rr} + t + 1. \end{array} \tag{52}$$

We define a *Non–Fixed Prototype* as a non–negative integer solution to those linear equations.

After calculating the prototypes, the backtrack search for finding the matrix C_A is similar to the backtrack algorithm for finding C. The only difference is that for C_A we should also consider the column sum which is equal to $s+1$.

After the matrix C_A is found, one can then apply a backtrack search to try to find the incidence matrix A of the partial geometry.

Note that the column orbits of the incidence matrix A are the orbits of the vertices of the line graph for the partial geometry. Thus, we only need to consider those column orbit sizes for which there exist an orbit matrix for the line graph with the corresponding orbit sizes.

Next, we consider an example.

Example 10. In this example, we consider the pg$(2, 2, 1)$. Its point graph is an srg$(15, 6, 1, 3)$. Assume $p = 3$ and assume that, for the point graph, there are three orbits of size one and four orbits of size three. Therefore $\varphi = 3$, $\psi = 4$, $n_1 = n_2 = n_3 = 1$ and $n_4 = n_5 = n_6 = n_7 = 3$. We have found the orbit matrix for the strongly regular graph with the same automorphism in Example 9, which is as follows up to isomorphism:

$$C = \left(\begin{array}{ccc|cccc} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 3 & 0 & 2 & 2 & 1 \\ 0 & 3 & 0 & 2 & 0 & 2 & 1 \\ 3 & 0 & 0 & 2 & 2 & 0 & 1 \\ 3 & 3 & 3 & 1 & 1 & 1 & 0 \end{array}\right).$$

Using Equation 45, we have:

$$C_{AA^T} = \left(\begin{array}{ccc|cccc} 3 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 3 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 3 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 3 & 3 & 2 & 2 & 1 \\ 0 & 3 & 0 & 2 & 3 & 2 & 1 \\ 3 & 0 & 0 & 2 & 2 & 3 & 1 \\ 3 & 3 & 3 & 1 & 1 & 1 & 3 \end{array}\right).$$

Now, we try to construct the orbit matrix of A. Assume that the desired partial geometry has 5 column orbits of size 3, that is, $m_1 = m_2 = m_3 = m_4 = m_5 = 3$. Therefore $\eta_1 = 0$ and $\eta_3 = 5$.

Now, we calculate the prototypes for C_A. First, we calculate the fixed prototype using Equation 51. We have the following set of equations:

$$\begin{array}{ccccccl} w_0 & + & w_1 & & & & = 0, \\ & & & z_0 & + & z_1 & = 5, \\ & & w_1 & & + & 3z_1 & = 3. \end{array} \tag{53}$$

The only non–negative integer solution to the set of equations above is:

$$(w_0, w_1, z_0, z_1) = (0, 0, 4, 1).$$

So there are exactly four 0's and one 1 on each fixed row of the matrix C_A.

Now, we consider the non–fixed prototypes, we have:

$$\begin{array}{ccccccccccl} w_0 & + & w_3 & & & & & & & & = 0, \\ & & & z_0 & + & z_1 & + & z_2 & + & z_3 & = 5, \\ & & w_3 & & + & z_1 & + & 2z_2 & + & 3z_3 & = 3, \\ & & 3w_3 & & + & z_1 & + & 4z_2 & + & 9z_3 & = c_{rr} + 3. \end{array} \tag{54}$$

In this example, for the non–fixed rows r, $c_{rr} = 0$. The only non–negative integer solution to the set of equations above is:

$$(w_0, w_3, z_0, z_1, z_2, z_3) = (0, 0, 2, 3, 0, 0).$$

So there are exactly two 0's and three 1's on each non–fixed row of the matrix C_A.

Now, we try to construct the matrix C_A using the information from the prototypes. The first row is [10000] up to isomorphism. Using Equation 49, we can see that

$$\sum_{i=1}^{5} u_{1i} u_{2i} = 0.$$

One can see, up to isomorphism, the second row is [01000]; the third row is [00100]; and so on. Therefore, the only possible choice for C_A up to isomorphism is the following:

$$C_A = \left(\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{array}\right).$$

Using this orbit matrix, a BDX backtrack algorithm finds the following partial geometry:

$$A = \left(\begin{array}{ccc|ccc|ccc|ccc|ccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ \hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right),$$

and

$$AA^T = \left(\begin{array}{c|c|c|ccc|ccc|ccc|ccc}
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline
0 & 3 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
0 & 0 & 3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
0 & 0 & 1 & 3 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 3 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline
0 & 1 & 0 & 1 & 1 & 0 & 3 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 3 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 3 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 3 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 3 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 1 \\ \hline
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 3 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 3
\end{array}\right).$$

Its point graph is:

$$B = \left(\begin{array}{c|c|c|ccc|ccc|ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}\right),$$

and

$$B^2 = \left(\begin{array}{c|c|c|ccc|ccc|ccc|ccc}
6 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline
3 & 6 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 3 & 3 & 3 & 1 & 1 & 1 \\ \hline
3 & 3 & 6 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 \\ \hline
3 & 3 & 1 & 6 & 3 & 3 & 1 & 3 & 1 & 1 & 1 & 3 & 3 & 3 & 1 \\
3 & 3 & 1 & 3 & 6 & 3 & 1 & 1 & 3 & 3 & 1 & 1 & 1 & 3 & 3 \\
3 & 3 & 1 & 3 & 3 & 6 & 3 & 1 & 1 & 1 & 3 & 1 & 3 & 1 & 3 \\ \hline
3 & 1 & 3 & 1 & 1 & 3 & 6 & 3 & 3 & 1 & 3 & 1 & 3 & 1 & 3 \\
3 & 1 & 3 & 3 & 1 & 1 & 3 & 6 & 3 & 1 & 1 & 3 & 3 & 3 & 1 \\
3 & 1 & 3 & 1 & 3 & 1 & 3 & 3 & 6 & 3 & 1 & 1 & 1 & 3 & 3 \\ \hline
1 & 3 & 3 & 1 & 3 & 1 & 1 & 1 & 3 & 6 & 3 & 3 & 1 & 3 & 3 \\
1 & 3 & 3 & 1 & 1 & 3 & 3 & 1 & 1 & 3 & 6 & 3 & 3 & 1 & 3 \\
1 & 3 & 3 & 3 & 1 & 1 & 1 & 3 & 1 & 3 & 3 & 6 & 3 & 3 & 1 \\ \hline
1 & 1 & 1 & 3 & 1 & 3 & 3 & 3 & 1 & 1 & 3 & 3 & 6 & 3 & 3 \\
1 & 1 & 1 & 3 & 3 & 1 & 1 & 3 & 3 & 3 & 1 & 3 & 3 & 6 & 3 \\
1 & 1 & 1 & 1 & 3 & 3 & 3 & 1 & 3 & 3 & 3 & 1 & 3 & 3 & 6
\end{array}\right).$$

□

5.4 Methodology

When searching for partial geometries, we assume that the partial geometry has an automorphism group of prime order p.

Depending on the parameters of the partial geometry, the construction process will be one of the following processes, or the combination of both.

Process 1:

1. Construct the orbit matrices of the point and/or line graph.
2. Construct the orbit matrices of the partial geometry from the orbit matrices of the point graph.
3. Construct the partial geometry from its orbit matrices.

Process 2:

1. Construct the orbit matrices of the point and/or line graph.

2. Construct the point and/or line graph.

3. Construct the partial geometry from its point and/or line graph.

We have implemented most of the programs required for Process 1 and Process 2.

5.5 Results

We have some preliminary results on pg(6, 10, 5). The point graph of a pg(6, 10, 5) is an srg(91, 66, 45, 55), which is the unique $T(14)$.

It can be seen that the automorphism group of $T(14)$ is the symmetric group S_{14}, since any permutation on the ground set keeps the graph unchanged. We are interested to know if $T(14)$ is geometric or not. This would show whether a pg(6, 10, 5) exists or does not exist. Using Lemma 2.30 it would be enough to check whether this graph is the point graph of a pls(6, 10) or not.

The prime factors of 14!, the size of $\mathrm{Aut}(T(14))$, are $2, 3, 5, 7, 11$, and 13. We have finished the search for $p = 5, 7, 11$ and 13, and no partial geometry was found for those group sizes. For $p = 11$ and $p = 13$ the computation time was low, about two hours CPU time, but for $p = 7$ and $p = 5$ the amount of computation was very large. We used more than 100 machines to finish the task in a reasonable amount of time.

Chapter 6

Conclusion

In this chapter we summaries the contributions of this thesis and talk about the possible future work.

6.1 Contribution

In this thesis, we studied the properties of strongly regular graphs and their automorphism groups.

The summary of the contributions in this thesis is as follows:

1. We implemented the theory of orbit matrices for strongly regular graphs for the first time which is a new theory in this field. We have implemented a computer program that generates all the orbit matrices for the given parameters and an automorphism.

2. Using orbit matrices we implemented an exhaustive search algorithm for finding strongly regular graphs. Using orbit matrices in our algorithm helps us to reduce the complexity as well as the required time of the algorithm.

3. Using our program we eliminated many primes as possible divisors of the order of the automorphism group of many unknown strongly regular graphs.

4. We found some new strongly regular graphs that are not isomorphic to any known ones.

5. We found several upper bounds on the number of fixed points ϕ of the automorphism of strongly regular graphs.

6. We have developed the theory of orbit matrices for partial geometries.

6.2 Future work

We have run the SRG program for several parameter sets, but because of time limitation, we were not able to do the search for other parameter sets. For future work, we can improve the program so that it can handle more parameter sets.

We have obtained some preliminary results on partial geometries. We can use the orbit matrix program for partial geometries to investigate their automorphisms and their existence.

We have found some upper bounds on the number of fixed points of an automorphism of a strongly regular graph. As future work, we can try to lower these upper bounds.

Bibliography

[1] L. BABAI, *Automorphism groups, isomorphism, reconstruction*, in Handbook of combinatorics, Vol. 1, 2, Elsevier, Amsterdam, 1995, pp. 1447–1540.

[2] R. C. BOSE, *Strongly regular graphs, partial geometries and partially balanced designs*, Pacific J. Math., 13 (1963), pp. 389–419.

[3] A. E. BROUWER AND J. H. VAN LINT, *Strongly regular graphs and partial geometries*, in Enumeration and design (Waterloo, Ont., 1982), Academic Press, Toronto, ON, 1984, pp. 85–122.

[4] R. A. BRUALDI AND H. J. RYSER, *Combinatorial matrix theory*, vol. 39 of Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, 1991.

[5] F. C. BUSSEMAKER, W. H. HAEMERS, R. MATHON, AND H. A. WILBRINK, *A $(49, 16, 3, 6)$ strongly regular graph does not exist*, European J. Combin., 10 (1989), pp. 413–418.

[6] P. J. CAMERON, *Automorphism groups of graphs*, in Selected topics in graph theory, 2, Academic Press, London, 1983, pp. 89–127.

[7] ——, *Random strongly regular graphs?*, Discrete Math., 273 (2003), pp. 103–114. EuroComb'01 (Barcelona).

[8] P. J. Cameron and J. H. van Lint, *Designs, graphs, codes and their links*, vol. 22 of London Mathematical Society Student Texts, Cambridge University Press, Cambridge, 1991.

[9] A. M. Cohen, *A new partial geometry with parameters* $(s, t, \alpha) = (7, 8, 4)$, J. Geom., 16 (1981), pp. 181–186.

[10] C. J. Colbourn and J. H. Dinitz, eds., *The CRC handbook of combinatorial designs*, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton, FL, 1996.

[11] ——, eds., *The CRC handbook of combinatorial designs*, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, 2006.

[12] J. B. Conway, *A course in functional analysis*, vol. 96 of Graduate Texts in Mathematics, Springer-Verlag, New York, second ed., 1990.

[13] K. Coolsaet, J. Degraer, and E. Spence, *The strongly regular* $(45, 12, 3, 3)$ *graphs*, Electron. J. Combin., 13 (2006), pp. Research Paper 32, 9 pp. (electronic).

[14] D. G. Corneil and R. A. Mathon, *Algorithmic techniques for the generation and analysis of strongly regular graphs and other combinatorial configurations*, Ann. Discrete Math., 2 (1978), pp. 1–32. Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976).

[15] J. Degraer and K. Coolsaet, *Classification of some strongly regular subgraphs of the McLaughlin graph*, Discrete Math., 308 (2008), pp. 395–400.

[16] P. Delsarte, J. M. Goethals, and J. J. Seidel, *Bounds for systems of lines and jacobi polynomials*, Philips Res. Repts., 30 (1975), pp. 91^*–95^*.

[17] J. Dénes and A. D. Keedwell, *Latin squares and their applications*. Academic Press, New York, 1974.

[18] P. Erdős and A. Rényi, *Asymmetric graphs*, Acta Math. Acad. Sci. Hungar, 14 (1963), pp. 295–315.

[19] D. T. Finkbeiner, II, *Introduction to matrices and linear transformations*, Second edition, W. H. Freeman and Co., San Francisco, Calif., 1966.

[20] A. Friedman, *Foundations of modern analysis*, Dover Publications Inc., New York, 1982. Reprint of the 1970 original.

[21] R. Frucht, *Herstellung von Graphen mit vorgegebener abstrakter Gruppe*, Compositio Math., 6 (1939), pp. 239–250.

[22] S. Geršgorin, *Über die abgrenzung der eigenwerte einer matrix*, Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk, 7 (1931), pp. 749–754.

[23] C. Godsil and G. Royle, *Algebraic graph theory*, vol. 207 of Graduate Texts in Mathematics, Springer-Verlag, New York, 2001.

[24] J. Hadamard, *Résolution d'une question relative aux déterminants*, Bull. des Sciences Math., 17 (1893), pp. 240–246.

[25] W. Haemers, *A new partial geometry constructed from the Hoffman-Singleton graph*, in Finite geometries and designs (Proc. Conf., Chelwood Gate, 1980), vol. 49 of London Math. Soc. Lecture Note Ser., Cambridge Univ. Press, Cambridge, 1981, pp. 119–127.

[26] W. H. Haemers and E. Spence, *The pseudo-geometric graphs for generalized quadrangles of order* $(3, t)$, European J. Combin., 22 (2001), pp. 839–845.

[27] K. Hoffman and R. Kunze, *Linear algebra*, Second edition, Prentice-Hall Inc., Englewood Cliffs, N.J., 1971.

[28] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[29] D. E. KNUTH, *Estimating the efficiency of backtrack programs*, Math. Comp., 29 (1975), pp. 122–136. Collection of articles dedicated to Derrick Henry Lehmer on the occasion of his seventieth birthday.

[30] D. L. KREHER AND D. R. STINSON, *Combinatorial algorithms: generation, enumeration, and search*, vol. 30, ACM, New York, NY, USA, 1999.

[31] M. G. KREĬN, *Hermitian-positive kernels in homogeneous spaces. II*, Ukrain. Nat. Žurnal, 2 (1950), pp. 10–59.

[32] C. LAM, *Computer construction of block designs*, in Surveys in combinatorics, 1997 (London), vol. 241 of London Math. Soc. Lecture Note Ser., Cambridge Univ. Press, Cambridge, 1997, pp. 49–64.

[33] S. LANG, *Linear algebra*, Second edition, Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1971.

[34] C. F. LAYWINE AND G. L. MULLEN, *Discrete mathematics using Latin squares*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons Inc., New York, 1998. A Wiley-Interscience Publication.

[35] A. A. MAKHNEV AND I. M. MINAKOV, *On automorphisms of strongly regular graphs with the parameters $\lambda = 1$ and $\mu = 2$*, Diskret. Mat., 16 (2004), pp. 95–104. Translation in Discrete Math. Appl. 14 (2004), no. 2, pp 201–210.

[36] R. MATHON, *On self-complementary strongly regular graphs*, Discrete Math., 69 (1988), pp. 263–281.

[37] ——, *Computational methods in design theory*, in Surveys in combinatorics, 1991 (Guildford, 1991), vol. 166 of London Math. Soc. Lecture Note Ser., Cambridge Univ. Press, Cambridge, 1991, pp. 101–117.

[38] B. D. MCKAY AND E. SPENCE, *Classification of regular two-graphs on 36 and 38 vertices*, Australas. J. Combin., 24 (2001), pp. 293–300.

[39] E. MENDELSOHN, *Every (finite) group is the group of automorphisms of a (finite) strongly regular graph*, Ars Combin., 6 (1978), pp. 75–86.

[40] D. V. PADUCHIKH, *On the automorphisms of strongly regular graphs with the parameters (85,14,3,2)*, Discrete Math. Appl., 19 (2009), pp. 89–111. Originally published in Diskretnaya Matematika (2009) 21, No. 1 (in Russian).

[41] D. V. PASECHNIK, *Skew-symmetric association schemes with two classes and strongly regular graphs of type* $L_{2n-1}(4n-1)$, Acta Appl. Math., 29 (1992), pp. 129–138. Interactions between algebra and combinatorics.

[42] D. J. S. ROBINSON, *A course in linear algebra with applications*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, second ed., 2006.

[43] L. L. SCOTT, *A condition on higman's parameters*, Notices Amer. Math. Soc., 20 (1973), pp. A–97.

[44] E. SPENCE, *The strongly regular* $(40, 12, 2, 4)$ *graphs*, Electron. J. Combin., 7 (2000), pp. Research Paper 22, 4 pp. (electronic).

[45] J. TITS, *Sur la trialité et certains groupes qui s'en déduisent*, Inst. Hautes Études Sci. Publ. Math., (1959), pp. 13–60.

[46] J. H. VAN LINT AND R. M. WILSON, *A course in combinatorics*, Cambridge University Press, Cambridge, second ed., 2001.

Appendix A

Search for unknown strongly regular graphs with less that 100 vertices

In this appendix, we provide the details of results of computer runs for strongly regular graphs. There is a table for each parameter set with fewer than one hundred vertices. The first row of each table corresponds to the smallest prime for which we were able to run the program. All subsequent rows correspond to other larger possible primes. The second column corresponds to the number of possible fixed points for each prime p in the first column. The third column corresponds to the number of orbit matrices found for the given number of fixed points. If there exist orbit matrices, then we run the SRG program for those orbit matrices; the fourth column shows the number of strongly regular graphs found for the orbit matrices. If there is a "?" in the table, it means that we have not found any solutions and we were not able to finish the program due to time constraints. Under the note column, we provide more information about the specific case. For example, if we have done an estimation for the particular case, it would be given in this column. We have found some upper bounds in this thesis on the number of fixed points an automorphism of a strongly regular graph can have. We refer to these upper bounds in the note column to shows why we do not need to check for bigger numbers of fixed points. If there is an "nnfp" in the note column, it means that there is no non–fixed prototype for that number of fixed points. By Theorem 3.4 there would be no more non–fixed prototypes for any larger number of fixed points.

p	#fix point	#orb matrix	#srg found	note
5	0	0		
	5	36	?	estimation 10 million days
	10	0		
	15	0		nnfp
7	2	0		
	9	0		nnfp
11	10	0		nnfp
13	0	0		
	13	0		nnfp
17	14	0		nnfp
19	8	0		nnfp
23	19	0		nnfp
29	7	0		nnfp
31	3	0		nnfp

Table 8: Computer run results on the automorphisms of srg(65, 32, 15, 16).

p	#fix point	#orb matrix	#srg found	note
5	4	27	0	
	9	1	0	
	14	0		
	19	0		
	24	0		
	29	0		nnfp
7	6	0		
	13	0		
	20	0		nnfp
11	3	0		
	14	0		nnfp
13	4	0		
	17	0		nnfp
17	1	0		
	18	0		nnfp
19	12	0		nnfp
23	0	2	0	

Table 9: Computer run results on the automorphisms of srg$(69, 20, 7, 5)$.

p	#fix point	#orb matrix	#srg found	note
5	0	231	0	
	5	0		
	10	0		
	15	0		
	20	0		nnfp
7	5	0		
	12	0		
	19	0		nnfp
11	9	0		nnfp
13	10	0		nnfp
17	7	0		nnfp
19	18	0		nnfp
23	6	0		nnfp
29	17	0		nnfp
31	13	0		nnfp

Table 10: Computer run results on the automorphisms of srg$(75, 32, 10, 16)$.

p	#fix point	#orb matrix	#srg found	note
5	1	0		
	6	0		
	11	0		
	16	0		
	21	0		nnfp
7	6	0		
	13	0		
	20	0		nnfp
11	10	0		nnfp
13	11	0		nnfp
17	8	0		nnfp
19	0	2	0	
	19	0		nnfp
23	7	0		nnfp
29	18	0		nnfp

Table 11: Computer run results on the automorphisms of srg(76, 30, 8, 14).

p	#fix point	#orb matrix	#srg found	note
5	1	4409	?	
	6	0		
	11	0		
	16	0		
	21	0		nnfp
7	6	0		
	13	0		
	20	0		nnfp
11	10	0		nnfp
13	11	0		nnfp
17	8	0		nnfp
19	0	1	0	
	19	0		nnfp
23	19	0		nnfp
29	18	0		nnfp
31	14	0		nnfp

Table 12: Computer run results on the automorphisms of srg(76, 35, 18, 14).

p	#fix point	#orb matrix	#srg found	note
3	1	0		
	4	2	0	
	7	0		
	10	0		
	13	0		
	16	0		
	19	0		
	22	0		
	25	0		
	28	0		Theorem 3.7
5	0	3	0	
	5	1	0	
	10	0		
	15	0		
	20	0		
	25	0		
	30	0		Theorem 3.7
7	1	8	0	
	8	0		
	15	0		
	22	0		
	29	0		Theorem 3.7
11	8	0		
	19	0		
	30	0		Theorem 3.7
13	7	0		
	20	0		
	33	0		Theorem 3.7
17	0	2	0	

Table 13: Computer run results on the automorphisms of srg(85, 14, 3, 2).

p	#fix point	#orb matrix	#srg found	note
5	0	>30000	?	
	5	236	?	
	10	3	?	estimation is 5×10^4 days for solution 1
	15	0		
	20	0		
	25	0		nnfp
7	1	0		
	8	0		
	15	0		
	22	0		nnfp
11	8	0		
	19	0		nnfp
13	7	0		
	20	0		nnfp
17	0	2	?	
	17	0		nnfp
19	9	0		nnfp
23	16	0		nnfp
29	27	0		nnfp

Table 14: Computer run results on the automorphisms of srg(85, 30, 11, 10).

p	#fix point	#orb matrix	#srg found	note
5	0	?		
	5	24994	?	
	10	0		
	15	0		nnfp
7	1	1536	?	
	8	0		
	15	0		nnfp
8	5	0		
	13			nnfp
9	4	0		
	13			nnfp
11	8	0		nnfp
13	7			nnfp
17	0	0		
	17	0		nnfp
19	9	0		nnfp
23	16	0		nnfp
29	27	0		nnfp
31	23	0		nnfp
37	11	0		nnfp
39	7	0		nnfp
41	3	0		nnfp

Table 15: Computer run results on the automorphisms of srg(85, 42, 20, 21).

p	#fix point	#orb matrix	#srg found	note
5	3	138	?	
	8	0		
	13	0		
	18	0		
	23	0		
	28	0		nnfp
7	4	0		
	11	0		
	18	0		
	25	0		nnfp
11	0	5	?	
	11	0		
	22	0		nnfp
13	10	0		
	23	0		nnfp
17	3	0		
	20	0		nnfp
19	12	0		nnfp
23	19	0		nnfp

Table 16: Computer run results on the automorphisms of srg$(88, 27, 6, 9)$.

p	#fix point	#orb matrix	#srg found	note
7	4	0		
	11	0		
	18	0		
	25			nnfp
11	7	0		
	18	0		nnfp
13	4	0		
	17	0		nnfp
17	10	0		nnfp
19	0	7	0	
	19	0		nnfp
23	3	0		
	26	0		nnfp
29	8	0		nnfp
31	2	0		
	33	0		nnfp
37	21	0		nnfp
39	17	0		nnfp

Table 17: Computer run results on the automorphisms of srg$(95, 40, 12, 20)$.

p	#fix point	#orb matrix	#srg found	note
5	1	?		
	6	0		
	11	0		
	16	0		
	21	0		
	26	0		nnfp
7	5	0		
	12	0		
	19	0		nnfp
11	8	0		
	19	0		nnfp
13	5	0		
	18	0		nnfp
17	11	0		nnfp
19	1	0		
	20	0		nnfp
23	4	0		
	27	0		nnfp
29	9	0		nnfp
31	3	0		nnfp

Table 18: Computer run results on the automorphisms of srg$(96, 35, 10, 14)$.

p	#fix point	#orb matrix	#srg found	note
7	5	0		
	12	0		
	19	0		
	26	0		nnfp
11	8	0		
	19	0		nnfp
13	5	0		
	18	0		nnfp
17	11	0		nnfp
19	1	8	0	
	20	0		nnfp
23	4	0		
	27	0		nnfp
29	9	0		nnfp
31	3	0		
	34	0		nnfp

Table 19: Computer run results on the automorphisms of srg(96, 38, 10, 18).

p	#fix point	#orb matrix	#srg found	note
7	5	0		
	12	0		
	19	0		
	26	0		nnfp
11	8	0		
	19	0		nnfp
13	5	0		
	18	0		nnfp
17	11	0		nnfp
19	1	0		
	20	0		nnfp
23	4	0		
	27	0		nnfp
29	9	0		nnfp
31	3	0		
	34	0		nnfp

Table 20: Computer run results on the automorphisms of srg(96, 45, 24, 18).

p	#fix point	#orb matrix	#srg found	note
3	0	?		
	3	0		
	6	0		
	9	0		
	12	0		
	15	0		
	18	0		
	21	0		Theorem 3.7
5	4	0		
	9	0		
	14	0		
	19	0		Theorem 3.7
7	1	0		
	8	0		
	15	0		
	22	0		Theorem 3.7
11	0	0		
	11	0		
	22	0		Theorem 3.7

Table 21: Computer run results on the automorphisms of srg$(99, 14, 1, 2)$.

p	#fix point	#orb matrix	#srg found	note
7	1	21989	?	
	8	0		
	15	0		
	22	0		
	29	0		nnfp
11	0	173	?	
	11	0		
	22	0		nnfp
13	8	0		
	21	0		nnfp
17	14	0		nnfp
19	4	0		
	23	0		nnfp
23	7	0		
	30	0		nnfp
29	12	0		nnfp
31	6	0		nnfp
37	25	0		nnfp
39	21	0		nnfp
41	17	0		nnfp

Table 22: Computer run results on the automorphisms of srg$(99, 42, 21, 15)$

Appendix B

srg(49, 18, 7, 6)

In this appendix, we provide the adjacency matrix of all srg(49, 18, 7, 6) that our SRG program has found. Table 23 shows a summery of all the graphs in this appendix.

Graph	Aut group size	From Latin square	New
A_1	10	Yes	No
A_2	15	Yes	No
A_3	30	No	Yes
A_4	15	No	Yes
A_5	15	No	Yes
A_6	21	No	Yes
A_7	1764	Yes	No
A_8	63	No	Yes
A_9	1008	Yes	No
A_{10}	126	No	Yes

Table 23: srg(49, 18, 7, 6) results summery

$$
A_1 = \left(\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
0 & 1 & 1 & 1 & 00000 & 00000 & 00000 & 00000 & 00000 & 00000 & 11111 & 11111 & 11111 \\ \hline
1 & 0 & 1 & 1 & 00000 & 00000 & 00000 & 00000 & 11111 & 11111 & 00000 & 00000 & 11111 \\ \hline
1 & 1 & 0 & 1 & 00000 & 00000 & 00000 & 11111 & 00000 & 11111 & 00000 & 11111 & 00000 \\ \hline
1 & 1 & 1 & 0 & 00000 & 00000 & 00000 & 11111 & 11111 & 00000 & 11111 & 00000 & 00000 \\ \hline
0 & 0 & 0 & 0 & 00000 & 01101 & 11001 & 11000 & 10001 & 01001 & 11000 & 10100 & 10010 \\
0 & 0 & 0 & 0 & 00000 & 10110 & 11100 & 01100 & 11000 & 10100 & 01100 & 01010 & 01001 \\
0 & 0 & 0 & 0 & 00000 & 01011 & 01110 & 00110 & 01100 & 01010 & 00110 & 00101 & 10100 \\
0 & 0 & 0 & 0 & 00000 & 10101 & 00111 & 00011 & 00110 & 00101 & 00011 & 10010 & 01010 \\
0 & 0 & 0 & 0 & 00000 & 11010 & 10011 & 10001 & 00011 & 10010 & 10001 & 01001 & 00101 \\ \hline
0 & 0 & 0 & 0 & 01011 & 00000 & 01011 & 11000 & 01100 & 10100 & 10010 & 11000 & 01100 \\
0 & 0 & 0 & 0 & 10101 & 00000 & 10101 & 01100 & 00110 & 01010 & 01001 & 01100 & 00110 \\
0 & 0 & 0 & 0 & 11010 & 00000 & 11010 & 00110 & 00011 & 00101 & 10100 & 00110 & 00011 \\
0 & 0 & 0 & 0 & 01101 & 00000 & 01101 & 00011 & 10001 & 10010 & 01010 & 00011 & 10001 \\
0 & 0 & 0 & 0 & 10110 & 00000 & 10110 & 10001 & 11000 & 01001 & 00101 & 10001 & 11000 \\ \hline
0 & 0 & 0 & 0 & 11001 & 01101 & 00000 & 10100 & 10010 & 00110 & 00101 & 11000 & 10001 \\
0 & 0 & 0 & 0 & 11100 & 10110 & 00000 & 01010 & 01001 & 00011 & 10010 & 01100 & 11000 \\
0 & 0 & 0 & 0 & 01110 & 01011 & 00000 & 00101 & 10100 & 10001 & 01001 & 00110 & 01100 \\
0 & 0 & 0 & 0 & 00111 & 10101 & 00000 & 10010 & 01010 & 11000 & 10100 & 00011 & 00110 \\
0 & 0 & 0 & 0 & 10011 & 11010 & 00000 & 01001 & 00101 & 01100 & 01010 & 10001 & 00011 \\ \hline
0 & 0 & 1 & 1 & 10001 & 10001 & 10010 & 01111 & 10000 & 10000 & 10000 & 10000 & 10100 \\
0 & 0 & 1 & 1 & 11000 & 11000 & 01001 & 10111 & 01000 & 01000 & 01000 & 01000 & 01010 \\
0 & 0 & 1 & 1 & 01100 & 01100 & 10100 & 11011 & 00100 & 00100 & 00100 & 00100 & 00101 \\
0 & 0 & 1 & 1 & 00110 & 00110 & 01010 & 11101 & 00010 & 00010 & 00010 & 00010 & 10010 \\
0 & 0 & 1 & 1 & 00011 & 00011 & 00101 & 11110 & 00001 & 00001 & 00001 & 00001 & 01001 \\ \hline
0 & 1 & 0 & 1 & 11000 & 00011 & 10100 & 10000 & 01111 & 10000 & 01000 & 10010 & 10000 \\
0 & 1 & 0 & 1 & 01100 & 10001 & 01010 & 01000 & 10111 & 01000 & 00100 & 01001 & 01000 \\
0 & 1 & 0 & 1 & 00110 & 11000 & 00101 & 00100 & 11011 & 00100 & 00010 & 10100 & 00100 \\
0 & 1 & 0 & 1 & 00011 & 01100 & 10010 & 00010 & 11101 & 00010 & 00001 & 01010 & 00010 \\
0 & 1 & 0 & 1 & 10001 & 00110 & 01001 & 00001 & 11110 & 00001 & 10000 & 00101 & 00001 \\ \hline
0 & 1 & 1 & 0 & 01001 & 10010 & 00110 & 10000 & 10000 & 01111 & 11000 & 00010 & 00100 \\
0 & 1 & 1 & 0 & 10100 & 01001 & 00011 & 01000 & 01000 & 10111 & 01100 & 00001 & 00010 \\
0 & 1 & 1 & 0 & 01010 & 10100 & 10001 & 00100 & 00100 & 11011 & 00110 & 10000 & 00001 \\
0 & 1 & 1 & 0 & 00101 & 01010 & 11000 & 00010 & 00010 & 11101 & 00011 & 01000 & 10000 \\
0 & 1 & 1 & 0 & 10010 & 00101 & 01100 & 00001 & 00001 & 11110 & 10001 & 00100 & 01000 \\ \hline
1 & 0 & 0 & 1 & 10001 & 10100 & 01010 & 10000 & 00001 & 10001 & 01111 & 00100 & 00100 \\
1 & 0 & 0 & 1 & 11000 & 01010 & 00101 & 01000 & 10000 & 11000 & 10111 & 00010 & 00010 \\
1 & 0 & 0 & 1 & 01100 & 00101 & 10010 & 00100 & 01000 & 01100 & 11011 & 00001 & 00001 \\
1 & 0 & 0 & 1 & 00110 & 10010 & 01001 & 00010 & 00100 & 00110 & 11101 & 10000 & 10000 \\
1 & 0 & 0 & 1 & 00011 & 01001 & 10100 & 00001 & 00010 & 00011 & 11110 & 01000 & 01000 \\ \hline
1 & 0 & 1 & 0 & 10010 & 10001 & 10001 & 10000 & 10100 & 00100 & 00010 & 01111 & 10000 \\
1 & 0 & 1 & 0 & 01001 & 11000 & 11000 & 01000 & 01010 & 00010 & 00001 & 10111 & 01000 \\
1 & 0 & 1 & 0 & 10100 & 01100 & 01100 & 00100 & 00101 & 00001 & 10000 & 11011 & 00100 \\
1 & 0 & 1 & 0 & 01010 & 00110 & 00110 & 00010 & 10010 & 10000 & 01000 & 11101 & 00010 \\
1 & 0 & 1 & 0 & 00101 & 00011 & 00011 & 00001 & 01001 & 01000 & 00100 & 11110 & 00001 \\ \hline
1 & 1 & 0 & 0 & 10100 & 00011 & 11000 & 10010 & 10000 & 00010 & 00010 & 10000 & 01111 \\
1 & 1 & 0 & 0 & 01010 & 10001 & 01100 & 01001 & 01000 & 00001 & 00001 & 01000 & 10111 \\
1 & 1 & 0 & 0 & 00101 & 11000 & 00110 & 10100 & 00100 & 10000 & 10000 & 00100 & 11011 \\
1 & 1 & 0 & 0 & 10010 & 01100 & 00011 & 01010 & 00010 & 01000 & 01000 & 00010 & 11101 \\
1 & 1 & 0 & 0 & 01001 & 00110 & 10001 & 00101 & 00001 & 00100 & 00100 & 00001 & 11110
\end{array}\right)
$$

autogp size = 10

Latin square graph. NOT NEW.

$$A_2 = \left(\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
0 & 1 & 1 & 1 & 00000 & 00000 & 00000 & 00000 & 00000 & 00000 & 11111 & 11111 & 11111 \\ \hline
1 & 0 & 1 & 1 & 00000 & 00000 & 00000 & 00000 & 11111 & 11111 & 00000 & 00000 & 11111 \\ \hline
1 & 1 & 0 & 1 & 00000 & 00000 & 00000 & 11111 & 00000 & 11111 & 00000 & 11111 & 00000 \\ \hline
1 & 1 & 1 & 0 & 00000 & 00000 & 00000 & 11111 & 11111 & 00000 & 11111 & 00000 & 00000 \\ \hline
0 & 0 & 0 & 0 & 00000 & 01011 & 10101 & 11000 & 11000 & 11000 & 10010 & 01100 & 10010 \\
0 & 0 & 0 & 0 & 00000 & 10101 & 11010 & 01100 & 01100 & 01100 & 01001 & 00110 & 01001 \\
0 & 0 & 0 & 0 & 00000 & 11010 & 01101 & 00110 & 00110 & 00110 & 10100 & 00011 & 10100 \\
0 & 0 & 0 & 0 & 00000 & 01101 & 10110 & 00011 & 00011 & 00011 & 01010 & 10001 & 01010 \\
0 & 0 & 0 & 0 & 00000 & 10110 & 01011 & 10001 & 10001 & 10001 & 00101 & 11000 & 00101 \\ \hline
0 & 0 & 0 & 0 & 01101 & 00000 & 01011 & 11000 & 10100 & 01010 & 11000 & 10001 & 01100 \\
0 & 0 & 0 & 0 & 10110 & 00000 & 10101 & 01100 & 01010 & 00101 & 01100 & 11000 & 00110 \\
0 & 0 & 0 & 0 & 01011 & 00000 & 11010 & 00110 & 00101 & 10010 & 00110 & 01100 & 00011 \\
0 & 0 & 0 & 0 & 10101 & 00000 & 01101 & 00011 & 10010 & 01001 & 00011 & 00110 & 10001 \\
0 & 0 & 0 & 0 & 11010 & 00000 & 10110 & 10001 & 01001 & 10100 & 10001 & 00011 & 11000 \\ \hline
0 & 0 & 0 & 0 & 11010 & 01101 & 00000 & 10100 & 00110 & 10001 & 11000 & 10100 & 10001 \\
0 & 0 & 0 & 0 & 01101 & 10110 & 00000 & 01010 & 00011 & 11000 & 01100 & 01010 & 11000 \\
0 & 0 & 0 & 0 & 10110 & 01011 & 00000 & 00101 & 10001 & 01100 & 00110 & 00101 & 01100 \\
0 & 0 & 0 & 0 & 01011 & 10101 & 00000 & 10010 & 11000 & 00110 & 00011 & 10010 & 00110 \\
0 & 0 & 0 & 0 & 10101 & 11010 & 00000 & 01001 & 01100 & 00011 & 10001 & 01001 & 00011 \\ \hline
0 & 0 & 1 & 1 & 10001 & 10001 & 10010 & 01111 & 10000 & 10000 & 10000 & 10000 & 10100 \\
0 & 0 & 1 & 1 & 11000 & 11000 & 01001 & 10111 & 01000 & 01000 & 01000 & 01000 & 01010 \\
0 & 0 & 1 & 1 & 01100 & 01100 & 10100 & 11011 & 00100 & 00100 & 00100 & 00100 & 00101 \\
0 & 0 & 1 & 1 & 00110 & 00110 & 01010 & 11101 & 00010 & 00010 & 00010 & 00010 & 10010 \\
0 & 0 & 1 & 1 & 00011 & 00011 & 00101 & 11110 & 00001 & 00001 & 00001 & 00001 & 01001 \\ \hline
0 & 1 & 0 & 1 & 10001 & 10010 & 00110 & 10000 & 01111 & 01000 & 00010 & 10100 & 00100 \\
0 & 1 & 0 & 1 & 11000 & 01001 & 00011 & 01000 & 10111 & 00100 & 00001 & 01010 & 00010 \\
0 & 1 & 0 & 1 & 01100 & 10100 & 10001 & 00100 & 11011 & 00010 & 10000 & 00101 & 00001 \\
0 & 1 & 0 & 1 & 00110 & 01010 & 11000 & 00010 & 11101 & 00001 & 01000 & 10010 & 10000 \\
0 & 1 & 0 & 1 & 00011 & 00101 & 01100 & 00001 & 11110 & 10000 & 00100 & 01001 & 01000 \\ \hline
0 & 1 & 1 & 0 & 10001 & 00101 & 11000 & 10000 & 00001 & 01111 & 10100 & 01000 & 10000 \\
0 & 1 & 1 & 0 & 11000 & 10010 & 01100 & 01000 & 10000 & 10111 & 01010 & 00100 & 01000 \\
0 & 1 & 1 & 0 & 01100 & 01001 & 00110 & 00100 & 01000 & 11011 & 00101 & 00010 & 00100 \\
0 & 1 & 1 & 0 & 00110 & 10100 & 00011 & 00010 & 00100 & 11101 & 10010 & 00001 & 00010 \\
0 & 1 & 1 & 0 & 00011 & 01010 & 10001 & 00001 & 00010 & 11110 & 01001 & 10000 & 00001 \\ \hline
1 & 0 & 0 & 1 & 10100 & 10001 & 10001 & 10000 & 00100 & 10010 & 01111 & 00001 & 10000 \\
1 & 0 & 0 & 1 & 01010 & 11000 & 11000 & 01000 & 00010 & 01001 & 10111 & 10000 & 01000 \\
1 & 0 & 0 & 1 & 00101 & 01100 & 01100 & 00100 & 00001 & 10100 & 11011 & 01000 & 00100 \\
1 & 0 & 0 & 1 & 10010 & 00110 & 00110 & 00010 & 10000 & 01010 & 11101 & 00100 & 00010 \\
1 & 0 & 0 & 1 & 01001 & 00011 & 00011 & 00001 & 01000 & 00101 & 11110 & 00010 & 00001 \\ \hline
1 & 0 & 1 & 0 & 00011 & 11000 & 10010 & 10000 & 10010 & 00001 & 01000 & 01111 & 00100 \\
1 & 0 & 1 & 0 & 10001 & 01100 & 01001 & 01000 & 01001 & 10000 & 00100 & 10111 & 00010 \\
1 & 0 & 1 & 0 & 11000 & 00110 & 10100 & 00100 & 10100 & 01000 & 00010 & 11011 & 00001 \\
1 & 0 & 1 & 0 & 01100 & 00011 & 01010 & 00010 & 01010 & 00100 & 00001 & 11101 & 10000 \\
1 & 0 & 1 & 0 & 00110 & 10001 & 00101 & 00001 & 00101 & 00010 & 10000 & 11110 & 01000 \\ \hline
1 & 1 & 0 & 0 & 10100 & 00011 & 11000 & 10010 & 00010 & 10000 & 10000 & 00010 & 01111 \\
1 & 1 & 0 & 0 & 01010 & 10001 & 01100 & 01001 & 00001 & 01000 & 01000 & 00001 & 10111 \\
1 & 1 & 0 & 0 & 00101 & 11000 & 00110 & 10100 & 10000 & 00100 & 00100 & 10000 & 11011 \\
1 & 1 & 0 & 0 & 10010 & 01100 & 00011 & 01010 & 01000 & 00010 & 00010 & 01000 & 11101 \\
1 & 1 & 0 & 0 & 01001 & 00110 & 10001 & 00101 & 00100 & 00001 & 00001 & 00100 & 11110
\end{array}\right)$$

autogp size = 15

Latin square graph. NOT NEW.

$A_3 =$

0	1	1	1	00000	00000	00000	00000	00000	00000	11111	11111	11111
1	0	1	1	00000	00000	00000	00000	11111	11111	00000	00000	11111
1	1	0	1	00000	00000	00000	11111	00000	11111	00000	11111	00000
1	1	1	0	00000	00000	00000	11111	11111	00000	11111	00000	00000
0	0	0	0	01111	10000	10000	10000	11000	11100	11010	10100	10000
0	0	0	0	10111	01000	01000	01000	01100	01110	01101	01010	01000
0	0	0	0	11011	00100	00100	00100	00110	00111	10110	00101	00100
0	0	0	0	11101	00010	00010	00010	00011	10011	01011	10010	00010
0	0	0	0	11110	00001	00001	00001	10001	11001	10101	01001	00001
0	0	0	0	10000	01111	10000	00110	11010	01000	00010	11100	01001
0	0	0	0	01000	10111	01000	00011	01101	00100	00001	01110	10100
0	0	0	0	00100	11011	00100	10001	10110	00010	10000	00111	01010
0	0	0	0	00010	11101	00010	11000	01011	00001	01000	10011	00101
0	0	0	0	00001	11110	00001	01100	10101	10000	00100	11001	10010
0	0	0	0	10000	10000	01111	10110	00010	10100	11000	01000	11001
0	0	0	0	01000	01000	10111	01011	00001	01010	01100	00100	11100
0	0	0	0	00100	00100	11011	10101	10000	00101	00110	00010	01110
0	0	0	0	00010	00010	11101	11010	01000	10010	00011	00001	00111
0	0	0	0	00001	00001	11110	01101	00100	01001	10001	10000	10011
0	0	1	1	10000	00110	10110	01001	11000	10100	11000	00011	00000
0	0	1	1	01000	00011	01011	10100	01100	01010	01100	10001	00000
0	0	1	1	00100	10001	10101	01010	00110	00101	00110	11000	00000
0	0	1	1	00010	11000	11010	00101	00011	10010	00011	01100	00000
0	0	1	1	00001	01100	01101	10010	10001	01001	10001	00110	00000
0	1	0	1	10001	10101	00100	10001	01001	11000	00110	00000	01010
0	1	0	1	11000	11010	00010	11000	10100	01100	00011	00000	00101
0	1	0	1	01100	01101	00001	01100	01010	00110	10001	00000	10010
0	1	0	1	00110	10110	10000	00110	00101	00011	11000	00000	01001
0	1	0	1	00011	01011	01000	00011	10010	10001	01100	00000	10100
0	1	1	0	10011	00001	10010	10010	10001	00110	00000	01001	10010
0	1	1	0	11001	10000	01001	01001	11000	00011	00000	10100	01001
0	1	1	0	11100	01000	10100	10100	01100	10001	00000	01010	10100
0	1	1	0	01110	00100	01010	01010	00110	11000	00000	00101	01010
0	1	1	0	00111	00010	00101	00101	00011	01100	00000	10010	00101
1	0	0	1	10101	00100	10001	10001	00110	00000	01001	00101	10001
1	0	0	1	11010	00010	11000	11000	00011	00000	10100	10010	11000
1	0	0	1	01101	00001	01100	01100	10001	00000	01010	01001	01100
1	0	0	1	10110	10000	00110	00110	11000	00000	00101	10100	00110
1	0	0	1	01011	01000	00011	00011	01100	00000	10010	01010	00011
1	0	1	0	10010	10011	00001	01100	00000	01001	01010	00110	10010
1	0	1	0	01001	11001	10000	00110	00000	10100	00101	00011	01001
1	0	1	0	10100	11100	01000	00011	00000	01010	10010	10001	10100
1	0	1	0	01010	01110	00100	10001	00000	00101	01001	11000	01010
1	0	1	0	00101	00111	00010	11000	00000	10010	10100	01100	00101
1	1	0	0	10000	01001	11001	00000	00101	10100	11000	10100	00110
1	1	0	0	01000	10100	11100	00000	10010	01010	01100	01010	00011
1	1	0	0	00100	01010	01110	00000	01001	00101	00110	00101	10001
1	1	0	0	00010	00101	00111	00000	10100	10010	00011	10010	11000
1	1	0	0	00001	10010	10011	00000	01010	01001	10001	01001	01100

autogp size = 30

NEW.

$A_4 =$

0	1	1	1	00000	00000	00000	00000	00000	00000	11111	11111	11111
1	0	1	1	00000	00000	00000	00000	11111	11111	00000	00000	11111
1	1	0	1	00000	00000	00000	11111	00000	11111	00000	11111	00000
1	1	1	0	00000	00000	00000	11111	11111	00000	11111	00000	00000
0	0	0	0	01111	10000	10000	10000	11000	11100	11010	10100	10000
0	0	0	0	10111	01000	01000	01000	01100	01110	01101	01010	01000
0	0	0	0	11011	00100	00100	00100	00110	00111	10110	00101	00100
0	0	0	0	11101	00010	00010	00010	00011	10011	01011	10010	00010
0	0	0	0	11110	00001	00001	00001	10001	11001	10101	01001	00001
0	0	0	0	10000	01111	01000	00011	11010	01000	00010	11100	10100
0	0	0	0	01000	10111	00100	10001	01101	00100	00001	01110	01010
0	0	0	0	00100	11011	00010	11000	10110	00010	10000	00111	00101
0	0	0	0	00010	11101	00001	01100	01011	00001	01000	10011	10010
0	0	0	0	00001	11110	10000	00110	10101	10000	00100	11001	01001
0	0	0	0	10000	00001	01111	10110	00100	10100	11000	10000	11001
0	0	0	0	01000	10000	10111	01011	00010	01010	01100	01000	11100
0	0	0	0	00100	01000	11011	10101	00001	00101	00110	00100	01110
0	0	0	0	00010	00100	11101	11010	10000	10010	00011	00010	00111
0	0	0	0	00001	00010	11110	01101	01000	01001	10001	00001	10011
0	0	1	1	10000	01100	10110	01001	10001	10100	11000	00110	00000
0	0	1	1	01000	00110	01011	10100	11000	01010	01100	00011	00000
0	0	1	1	00100	00011	10101	01010	01100	00101	00110	10001	00000
0	0	1	1	00010	10001	11010	00101	00110	10010	00011	11000	00000
0	0	1	1	00001	11000	01101	10010	00011	01001	10001	01100	00000
0	1	0	1	10001	10101	00010	11000	01001	11000	00110	00000	00101
0	1	0	1	11000	11010	00001	01100	10100	01100	00011	00000	10010
0	1	0	1	01100	01101	10000	00110	01010	00110	10001	00000	01001
0	1	0	1	00110	10110	01000	00011	00101	00011	11000	00000	10100
0	1	0	1	00011	01011	00100	10001	10010	10001	01100	00000	01010
0	1	1	0	10011	00001	10010	10010	10001	00110	00000	01001	10010
0	1	1	0	11001	10000	01001	01001	11000	00011	00000	10100	01001
0	1	1	0	11100	01000	10100	10100	01100	10001	00000	01010	10100
0	1	1	0	01110	00100	01010	01010	00110	11000	00000	00101	01010
0	1	1	0	00111	00010	00101	00101	00011	01100	00000	10010	00101
1	0	0	1	10101	00100	10001	10001	00110	00000	01001	00101	10001
1	0	0	1	11010	00010	11000	11000	00011	00000	10100	10010	11000
1	0	0	1	01101	00001	01100	01100	10001	00000	01010	01001	01100
1	0	0	1	10110	10000	00110	00110	11000	00000	00101	10100	00110
1	0	0	1	01011	01000	00011	00011	01100	00000	10010	01010	00011
1	0	1	0	10010	10011	10000	00110	00000	01001	01010	00110	01001
1	0	1	0	01001	11001	01000	00011	00000	10100	00101	00011	10100
1	0	1	0	10100	11100	00100	10001	00000	01010	10010	10001	01010
1	0	1	0	01010	01110	00010	11000	00000	00101	01001	11000	00101
1	0	1	0	00101	00111	00001	01100	00000	10010	10100	01100	10010
1	1	0	0	10000	10010	11001	00000	01010	10100	11000	01001	00110
1	1	0	0	01000	01001	11100	00000	00101	01010	01100	10100	00011
1	1	0	0	00100	10100	01110	00000	10010	00101	00110	01010	10001
1	1	0	0	00010	01010	00111	00000	01001	10010	00011	00101	11000
1	1	0	0	00001	00101	10011	00000	10100	01001	10001	10010	01100

autogp size = 15

NEW.

$$
A_5 = \left(\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
0 & 1 & 1 & 1 & 00000 & 00000 & 00000 & 00000 & 00000 & 00000 & 11111 & 11111 & 11111 \\ \hline
1 & 0 & 1 & 1 & 00000 & 00000 & 00000 & 00000 & 11111 & 11111 & 00000 & 00000 & 11111 \\ \hline
1 & 1 & 0 & 1 & 00000 & 00000 & 00000 & 11111 & 00000 & 11111 & 00000 & 11111 & 00000 \\ \hline
1 & 1 & 1 & 0 & 00000 & 00000 & 00000 & 11111 & 11111 & 00000 & 11111 & 00000 & 00000 \\ \hline
0 & 0 & 0 & 0 & 01111 & 10000 & 10000 & 10000 & 11000 & 11100 & 11010 & 10100 & 10000 \\
0 & 0 & 0 & 0 & 10111 & 01000 & 01000 & 01000 & 01100 & 01110 & 01101 & 01010 & 01000 \\
0 & 0 & 0 & 0 & 11011 & 00100 & 00100 & 00100 & 00110 & 00111 & 10110 & 00101 & 00100 \\
0 & 0 & 0 & 0 & 11101 & 00010 & 00010 & 00010 & 00011 & 10011 & 01011 & 10010 & 00010 \\
0 & 0 & 0 & 0 & 11110 & 00001 & 00001 & 00001 & 10001 & 11001 & 10101 & 01001 & 00001 \\ \hline
0 & 0 & 0 & 0 & 10000 & 01111 & 01000 & 10100 & 11010 & 01000 & 00010 & 11100 & 00011 \\
0 & 0 & 0 & 0 & 01000 & 10111 & 00100 & 01010 & 01101 & 00100 & 00001 & 01110 & 10001 \\
0 & 0 & 0 & 0 & 00100 & 11011 & 00010 & 00101 & 10110 & 00010 & 10000 & 00111 & 11000 \\
0 & 0 & 0 & 0 & 00010 & 11101 & 00001 & 10010 & 01011 & 00001 & 01000 & 10011 & 01100 \\
0 & 0 & 0 & 0 & 00001 & 11110 & 10000 & 01001 & 10101 & 10000 & 00100 & 11001 & 00110 \\ \hline
0 & 0 & 0 & 0 & 10000 & 00001 & 01111 & 11001 & 00100 & 10100 & 11000 & 10000 & 10110 \\
0 & 0 & 0 & 0 & 01000 & 10000 & 10111 & 11100 & 00010 & 01010 & 01100 & 01000 & 01011 \\
0 & 0 & 0 & 0 & 00100 & 01000 & 11011 & 01110 & 00001 & 00101 & 00110 & 00100 & 10101 \\
0 & 0 & 0 & 0 & 00010 & 00100 & 11101 & 00111 & 10000 & 10010 & 00011 & 00010 & 11010 \\
0 & 0 & 0 & 0 & 00001 & 00010 & 11110 & 10011 & 01000 & 01001 & 10001 & 00001 & 01101 \\ \hline
0 & 0 & 1 & 1 & 10000 & 10010 & 11001 & 00110 & 01010 & 10100 & 11000 & 01001 & 00000 \\
0 & 0 & 1 & 1 & 01000 & 01001 & 11100 & 00011 & 00101 & 01010 & 01100 & 10100 & 00000 \\
0 & 0 & 1 & 1 & 00100 & 10100 & 01110 & 10001 & 10010 & 00101 & 00110 & 01010 & 00000 \\
0 & 0 & 1 & 1 & 00010 & 01010 & 00111 & 11000 & 01001 & 10010 & 00011 & 00101 & 00000 \\
0 & 0 & 1 & 1 & 00001 & 00101 & 10011 & 01100 & 10100 & 01001 & 10001 & 10010 & 00000 \\ \hline
0 & 1 & 0 & 1 & 10001 & 10101 & 00010 & 00101 & 01001 & 11000 & 00110 & 00000 & 11000 \\
0 & 1 & 0 & 1 & 11000 & 11010 & 00001 & 10010 & 10100 & 01100 & 00011 & 00000 & 01100 \\
0 & 1 & 0 & 1 & 01100 & 01101 & 10000 & 01001 & 01010 & 00110 & 10001 & 00000 & 00110 \\
0 & 1 & 0 & 1 & 00110 & 10110 & 01000 & 10100 & 00101 & 00011 & 11000 & 00000 & 00011 \\
0 & 1 & 0 & 1 & 00011 & 01011 & 00100 & 01010 & 10010 & 10001 & 01100 & 00000 & 10001 \\ \hline
0 & 1 & 1 & 0 & 10011 & 00001 & 10010 & 10010 & 10001 & 00110 & 00000 & 01001 & 10010 \\
0 & 1 & 1 & 0 & 11001 & 10000 & 01001 & 01001 & 11000 & 00011 & 00000 & 10100 & 01001 \\
0 & 1 & 1 & 0 & 11100 & 01000 & 10100 & 10100 & 01100 & 10001 & 00000 & 01010 & 10100 \\
0 & 1 & 1 & 0 & 01110 & 00100 & 01010 & 01010 & 00110 & 11000 & 00000 & 00101 & 01010 \\
0 & 1 & 1 & 0 & 00111 & 00010 & 00101 & 00101 & 00011 & 01100 & 00000 & 10010 & 00101 \\ \hline
1 & 0 & 0 & 1 & 10101 & 00100 & 10001 & 10001 & 00110 & 00000 & 01001 & 00101 & 10001 \\
1 & 0 & 0 & 1 & 11010 & 00010 & 11000 & 11000 & 00011 & 00000 & 10100 & 10010 & 11000 \\
1 & 0 & 0 & 1 & 01101 & 00001 & 01100 & 01100 & 10001 & 00000 & 01010 & 01001 & 01100 \\
1 & 0 & 0 & 1 & 10110 & 10000 & 00110 & 00110 & 11000 & 00000 & 00101 & 10100 & 00110 \\
1 & 0 & 0 & 1 & 01011 & 01000 & 00011 & 00011 & 01100 & 00000 & 10010 & 01010 & 00011 \\ \hline
1 & 0 & 1 & 0 & 10010 & 10011 & 10000 & 01001 & 00000 & 01001 & 01010 & 00110 & 00110 \\
1 & 0 & 1 & 0 & 01001 & 11001 & 01000 & 10100 & 00000 & 10100 & 00101 & 00011 & 00011 \\
1 & 0 & 1 & 0 & 10100 & 11100 & 00100 & 01010 & 00000 & 01010 & 10010 & 10001 & 10001 \\
1 & 0 & 1 & 0 & 01010 & 01110 & 00010 & 00101 & 00000 & 00101 & 01001 & 11000 & 11000 \\
1 & 0 & 1 & 0 & 00101 & 00111 & 00001 & 10010 & 00000 & 10010 & 10100 & 01100 & 01100 \\ \hline
1 & 1 & 0 & 0 & 10000 & 01100 & 10110 & 00000 & 10001 & 10100 & 11000 & 00110 & 01001 \\
1 & 1 & 0 & 0 & 01000 & 00110 & 01011 & 00000 & 11000 & 01010 & 01100 & 00011 & 10100 \\
1 & 1 & 0 & 0 & 00100 & 00011 & 10101 & 00000 & 01100 & 00101 & 00110 & 10001 & 01010 \\
1 & 1 & 0 & 0 & 00010 & 10001 & 11010 & 00000 & 00110 & 10010 & 00011 & 11000 & 00101 \\
1 & 1 & 0 & 0 & 00001 & 11000 & 01101 & 00000 & 00011 & 01001 & 10001 & 01100 & 10010
\end{array}\right)
$$

autogp size = 15

NEW.

$$
A_6 = \left(\begin{array}{c|c|c|c|c|c|c}
0000000 & 1110000 & 1110000 & 1100100 & 1100100 & 1010100 & 1010100 \\
0000000 & 0111000 & 0111000 & 0110010 & 0110010 & 0101010 & 0101010 \\
0000000 & 0011100 & 0011100 & 0011001 & 0011001 & 0010101 & 0010101 \\
0000000 & 0001110 & 0001110 & 1001100 & 1001100 & 1001010 & 1001010 \\
0000000 & 0000111 & 0000111 & 0100110 & 0100110 & 0100101 & 0100101 \\
0000000 & 1000011 & 1000011 & 0010011 & 0010011 & 1010010 & 1010010 \\
0000000 & 1100001 & 1100001 & 1001001 & 1001001 & 0101001 & 0101001 \\
\hline
1000011 & 0000000 & 1100100 & 1100010 & 0001011 & 1010010 & 0001101 \\
1100001 & 0000000 & 0110010 & 0110001 & 1000101 & 0101001 & 1000110 \\
1110000 & 0000000 & 0011001 & 1011000 & 1100010 & 1010100 & 0100011 \\
0111000 & 0000000 & 1001100 & 0101100 & 0110001 & 0101010 & 1010001 \\
0011100 & 0000000 & 0100110 & 0010110 & 1011000 & 0010101 & 1101000 \\
0001110 & 0000000 & 0010011 & 0001011 & 0101100 & 1001010 & 0110100 \\
0000111 & 0000000 & 1001001 & 1000101 & 0010110 & 0100101 & 0011010 \\
\hline
1000011 & 1001001 & 0000000 & 1010100 & 0101001 & 0100110 & 1110000 \\
1100001 & 1100100 & 0000000 & 0101010 & 1010100 & 0010011 & 0111000 \\
1110000 & 0110010 & 0000000 & 0010101 & 0101010 & 1001001 & 0011100 \\
0111000 & 0011001 & 0000000 & 1001010 & 0010101 & 1100100 & 0001110 \\
0011100 & 1001100 & 0000000 & 0100101 & 1001010 & 0110010 & 0000111 \\
0001110 & 0100110 & 0000000 & 1010010 & 0100101 & 0011001 & 1000011 \\
0000111 & 0010011 & 0000000 & 0101001 & 1010010 & 1001100 & 1100001 \\
\hline
1001001 & 1010001 & 1001010 & 0000000 & 0101100 & 0011100 & 0001011 \\
1100100 & 1101000 & 0100101 & 0000000 & 0010110 & 0001110 & 1000101 \\
0110010 & 0110100 & 1010010 & 0000000 & 0001011 & 0000111 & 1100010 \\
0011001 & 0011010 & 0101001 & 0000000 & 1000101 & 1000011 & 0110001 \\
1001100 & 0001101 & 1010100 & 0000000 & 1100010 & 1100001 & 1011000 \\
0100110 & 1000110 & 0101010 & 0000000 & 0110001 & 1110000 & 0101100 \\
0010011 & 0100011 & 0010101 & 0000000 & 1011000 & 0111000 & 0010110 \\
\hline
1001001 & 0110100 & 0100101 & 0001101 & 0000000 & 1110000 & 1100010 \\
1100100 & 0011010 & 1010010 & 1000110 & 0000000 & 0111000 & 0110001 \\
0110010 & 0001101 & 0101001 & 0100011 & 0000000 & 0011100 & 1011000 \\
0011001 & 1000110 & 1010100 & 1010001 & 0000000 & 0001110 & 0101100 \\
1001100 & 0100011 & 0101010 & 1101000 & 0000000 & 0000111 & 0010110 \\
0100110 & 1010001 & 0010101 & 0110100 & 0000000 & 1000011 & 0001011 \\
0010011 & 1101000 & 1001010 & 0011010 & 0000000 & 1100001 & 1000101 \\
\hline
1001010 & 1010010 & 0011001 & 0001110 & 1000011 & 0000000 & 1001100 \\
0100101 & 0101001 & 1001100 & 0000111 & 1100001 & 0000000 & 0100110 \\
1010010 & 1010100 & 0100110 & 1000011 & 1110000 & 0000000 & 0010011 \\
0101001 & 0101010 & 0010011 & 1100001 & 0111000 & 0000000 & 1001001 \\
1010100 & 0010101 & 1001001 & 1110000 & 0011100 & 0000000 & 1100100 \\
0101010 & 1001010 & 1100100 & 0111000 & 0001110 & 0000000 & 0110010 \\
0010101 & 0100101 & 0110010 & 0011100 & 0000111 & 0000000 & 0011001 \\
\hline
1001010 & 0101100 & 1000011 & 0110100 & 1010001 & 1001100 & 0000000 \\
0100101 & 0010110 & 1100001 & 0011010 & 1101000 & 0100110 & 0000000 \\
1010010 & 0001011 & 1110000 & 0001101 & 0110100 & 0010011 & 0000000 \\
0101001 & 1000101 & 0111000 & 1000110 & 0011010 & 1001001 & 0000000 \\
1010100 & 1100010 & 0011100 & 0100011 & 0001101 & 1100100 & 0000000 \\
0101010 & 0110001 & 0001110 & 1010001 & 1000110 & 0110010 & 0000000 \\
0010101 & 1011000 & 0000111 & 1101000 & 0100011 & 0011001 & 0000000
\end{array}\right)
$$

autogp size = 21

NEW.

$$
A_7 = \left(\begin{array}{c|c|c|c|c|c|c}
0000000 & 1110000 & 1110000 & 1100100 & 1100100 & 1010100 & 1010100 \\
0000000 & 0111000 & 0111000 & 0110010 & 0110010 & 0101010 & 0101010 \\
0000000 & 0011100 & 0011100 & 0011001 & 0011001 & 0010101 & 0010101 \\
0000000 & 0001110 & 0001110 & 1001100 & 1001100 & 1001010 & 1001010 \\
0000000 & 0000111 & 0000111 & 0100110 & 0100110 & 0100101 & 0100101 \\
0000000 & 1000011 & 1000011 & 0010011 & 0010011 & 1010010 & 1010010 \\
0000000 & 1100001 & 1100001 & 1001001 & 1001001 & 0101001 & 0101001 \\
\hline
1000011 & 0000000 & 1010010 & 1001001 & 0101010 & 1110000 & 0100110 \\
1100001 & 0000000 & 0101001 & 1100100 & 0010101 & 0111000 & 0010011 \\
1110000 & 0000000 & 1010100 & 0110010 & 1001010 & 0011100 & 1001001 \\
0111000 & 0000000 & 0101010 & 0011001 & 0100101 & 0001110 & 1100100 \\
0011100 & 0000000 & 0010101 & 1001100 & 1010010 & 0000111 & 0110010 \\
0001110 & 0000000 & 1001010 & 0100110 & 0101001 & 1000011 & 0011001 \\
0000111 & 0000000 & 0100101 & 0010011 & 1010100 & 1100001 & 1001100 \\
\hline
1000011 & 1010010 & 0000000 & 0101010 & 1001001 & 0100110 & 1110000 \\
1100001 & 0101001 & 0000000 & 0010101 & 1100100 & 0010011 & 0111000 \\
1110000 & 1010100 & 0000000 & 1001010 & 0110010 & 1001001 & 0011100 \\
0111000 & 0101010 & 0000000 & 0100101 & 0011001 & 1100100 & 0001110 \\
0011100 & 0010101 & 0000000 & 1010010 & 1001100 & 0110010 & 0000111 \\
0001110 & 1001010 & 0000000 & 0101001 & 0100110 & 0011001 & 1000011 \\
0000111 & 0100101 & 0000000 & 1010100 & 0010011 & 1001100 & 1100001 \\
\hline
1001001 & 1100100 & 0010101 & 0000000 & 1100001 & 1001010 & 0000111 \\
1100100 & 0110010 & 1001010 & 0000000 & 1110000 & 0100101 & 1000011 \\
0110010 & 0011001 & 0100101 & 0000000 & 0111000 & 1010010 & 1100001 \\
0011001 & 1001100 & 1010010 & 0000000 & 0011100 & 0101001 & 1110000 \\
1001100 & 0100110 & 0101001 & 0000000 & 0001110 & 1010100 & 0111000 \\
0100110 & 0010011 & 1010100 & 0000000 & 0000111 & 0101010 & 0011100 \\
0010011 & 1001001 & 0101010 & 0000000 & 1000011 & 0010101 & 0001110 \\
\hline
1001001 & 0010101 & 1100100 & 1100001 & 0000000 & 0000111 & 1001010 \\
1100100 & 1001010 & 0110010 & 1110000 & 0000000 & 1000011 & 0100101 \\
0110010 & 0100101 & 0011001 & 0111000 & 0000000 & 1100001 & 1010010 \\
0011001 & 1010010 & 1001100 & 0011100 & 0000000 & 1110000 & 0101001 \\
1001100 & 0101001 & 0100110 & 0001110 & 0000000 & 0111000 & 1010100 \\
0100110 & 1010100 & 0010011 & 0000111 & 0000000 & 0011100 & 0101010 \\
0010011 & 0101010 & 1001001 & 1000011 & 0000000 & 0001110 & 0010101 \\
\hline
1001010 & 1000011 & 0011001 & 1010100 & 0111000 & 0000000 & 1001100 \\
0100101 & 1100001 & 1001100 & 0101010 & 0011100 & 0000000 & 0100110 \\
1010010 & 1110000 & 0100110 & 0010101 & 0001110 & 0000000 & 0010011 \\
0101001 & 0111000 & 0010011 & 1001010 & 0000111 & 0000000 & 1001001 \\
1010100 & 0011100 & 1001001 & 0100101 & 1000011 & 0000000 & 1100100 \\
0101010 & 0001110 & 1100100 & 1010010 & 1100001 & 0000000 & 0110010 \\
0010101 & 0000111 & 0110010 & 0101001 & 1110000 & 0000000 & 0011001 \\
\hline
1001010 & 0011001 & 1000011 & 0111000 & 1010100 & 1001100 & 0000000 \\
0100101 & 1001100 & 1100001 & 0011100 & 0101010 & 0100110 & 0000000 \\
1010010 & 0100110 & 1110000 & 0001110 & 0010101 & 0010011 & 0000000 \\
0101001 & 0010011 & 0111000 & 0000111 & 1001010 & 1001001 & 0000000 \\
1010100 & 1001001 & 0011100 & 1000011 & 0100101 & 1100100 & 0000000 \\
0101010 & 1100100 & 0001110 & 1100001 & 1010010 & 0110010 & 0000000 \\
0010101 & 0110010 & 0000111 & 1110000 & 0101001 & 0011001 & 0000000
\end{array}\right)
$$

autogp size = 1764

Latin square graph, NOT NEW.

$$A_8 = \left(\begin{array}{c|c|c|c|c|c|c}
0000000 & 1110000 & 1101000 & 1101000 & 1100100 & 1100010 & 1010100 \\
0000000 & 0111000 & 0110100 & 0110100 & 0110010 & 0110001 & 0101010 \\
0000000 & 0011100 & 0011010 & 0011010 & 0011001 & 1011000 & 0010101 \\
0000000 & 0001110 & 0001101 & 0001101 & 1001100 & 0101100 & 1001010 \\
0000000 & 0000111 & 1000110 & 1000110 & 0100110 & 0010110 & 0100101 \\
0000000 & 1000011 & 0100011 & 0100011 & 0010011 & 0001011 & 1010010 \\
0000000 & 1100001 & 1010001 & 1010001 & 1001001 & 1000101 & 0101001 \\
\hline
1000011 & 0000000 & 1100100 & 0110001 & 0001101 & 0101010 & 0110001 \\
1100001 & 0000000 & 0110010 & 1011000 & 1000110 & 0010101 & 1011000 \\
1110000 & 0000000 & 0011001 & 0101100 & 0100011 & 1001010 & 0101100 \\
0111000 & 0000000 & 1001100 & 0010110 & 1010001 & 0100101 & 0010110 \\
0011100 & 0000000 & 0100110 & 0001011 & 1101000 & 1010010 & 0001011 \\
0001110 & 0000000 & 0010011 & 1000101 & 0110100 & 0101001 & 1000101 \\
0000111 & 0000000 & 1001001 & 1100010 & 0011010 & 1010100 & 1100010 \\
\hline
1000101 & 1001001 & 0000000 & 0110100 & 1010100 & 1000110 & 0000111 \\
1100010 & 1100100 & 0000000 & 0011010 & 0101010 & 0100011 & 1000011 \\
0110001 & 0110010 & 0000000 & 0001101 & 0010101 & 1010001 & 1100001 \\
1011000 & 0011001 & 0000000 & 1000110 & 1001010 & 1101000 & 1110000 \\
0101100 & 1001100 & 0000000 & 0100011 & 0100101 & 0110100 & 0111000 \\
0010110 & 0100110 & 0000000 & 1010001 & 1010010 & 0011010 & 0011100 \\
0001011 & 0010011 & 0000000 & 1101000 & 0101001 & 0001101 & 0001110 \\
\hline
1000101 & 0100011 & 0001011 & 0000000 & 1101000 & 0110001 & 0110100 \\
1100010 & 1010001 & 1000101 & 0000000 & 0110100 & 1011000 & 0011010 \\
0110001 & 1101000 & 1100010 & 0000000 & 0011010 & 0101100 & 0001101 \\
1011000 & 0110100 & 0110001 & 0000000 & 0001101 & 0010110 & 1000110 \\
0101100 & 0011010 & 1011000 & 0000000 & 1000110 & 0001011 & 0100011 \\
0010110 & 0001101 & 0101100 & 0000000 & 0100011 & 1000101 & 1010001 \\
0001011 & 1000110 & 0010110 & 0000000 & 1010001 & 1100010 & 1101000 \\
\hline
1001001 & 0101100 & 1001010 & 1000101 & 0000000 & 1000011 & 0011010 \\
1100100 & 0010110 & 0100101 & 1100010 & 0000000 & 1100001 & 0001101 \\
0110010 & 0001011 & 1010010 & 0110001 & 0000000 & 1110000 & 1000110 \\
0011001 & 1000101 & 0101001 & 1011000 & 0000000 & 0111000 & 0100011 \\
1001100 & 1100010 & 1010100 & 0101100 & 0000000 & 0011100 & 1010001 \\
0100110 & 0110001 & 0101010 & 0010110 & 0000000 & 0001110 & 1101000 \\
0010011 & 1011000 & 0010101 & 0001011 & 0000000 & 0000111 & 0110100 \\
\hline
1010001 & 0010101 & 1011000 & 0100011 & 1110000 & 0000000 & 1001001 \\
1101000 & 1001010 & 0101100 & 1010001 & 0111000 & 0000000 & 1100100 \\
0110100 & 0100101 & 0010110 & 1101000 & 0011100 & 0000000 & 0110010 \\
0011010 & 1010010 & 0001011 & 0110100 & 0001110 & 0000000 & 0011001 \\
0001101 & 0101001 & 1000101 & 0011010 & 0000111 & 0000000 & 1001100 \\
1000110 & 1010100 & 1100010 & 0001101 & 1000011 & 0000000 & 0100110 \\
0100011 & 0101010 & 0110001 & 1000110 & 1100001 & 0000000 & 0010011 \\
\hline
1001010 & 0100011 & 0111000 & 0001011 & 0010110 & 1100100 & 0000000 \\
0100101 & 1010001 & 0011100 & 1000101 & 0001011 & 0110010 & 0000000 \\
1010010 & 1101000 & 0001110 & 1100010 & 1000101 & 0011001 & 0000000 \\
0101001 & 0110100 & 0000111 & 0110001 & 1100010 & 1001100 & 0000000 \\
1010100 & 0011010 & 1000011 & 1011000 & 0110001 & 0100110 & 0000000 \\
0101010 & 0001101 & 1100001 & 0101100 & 1011000 & 0010011 & 0000000 \\
0010101 & 1000110 & 1110000 & 0010110 & 0101100 & 1001001 & 0000000 \\
\hline
\end{array}\right)$$

autogp size = 63

NEW.

$$A_9 = \left(\begin{array}{c|c|c|c|c|c|c}
0000000 & 1110000 & 1101000 & 1101000 & 1100100 & 1100010 & 1010100 \\
0000000 & 0111000 & 0110100 & 0110100 & 0110010 & 0110001 & 0101010 \\
0000000 & 0011100 & 0011010 & 0011010 & 0011001 & 1011000 & 0010101 \\
0000000 & 0001110 & 0001101 & 0001101 & 1001100 & 0101100 & 1001010 \\
0000000 & 0000111 & 1000110 & 1000110 & 0100110 & 0010110 & 0100101 \\
0000000 & 1000011 & 0100011 & 0100011 & 0010011 & 0001011 & 1010010 \\
0000000 & 1100001 & 1010001 & 1010001 & 1001001 & 1000101 & 0101001 \\
\hline
1000011 & 0000000 & 1010001 & 1001010 & 0110001 & 0100110 & 0011010 \\
1100001 & 0000000 & 1101000 & 0100101 & 1011000 & 0010011 & 0001101 \\
1110000 & 0000000 & 0110100 & 1010010 & 0101100 & 1001001 & 1000110 \\
0111000 & 0000000 & 0011010 & 0101001 & 0010110 & 1100100 & 0100011 \\
0011100 & 0000000 & 0001101 & 1010100 & 0001011 & 0110010 & 1010001 \\
0001110 & 0000000 & 1000110 & 0101010 & 1000101 & 0011001 & 1101000 \\
0000111 & 0000000 & 0100011 & 0010101 & 1100010 & 1001100 & 0110100 \\
\hline
1000101 & 1100010 & 0000000 & 0001011 & 0101100 & 0110001 & 0110001 \\
1100010 & 0110001 & 0000000 & 1000101 & 0010110 & 1011000 & 1011000 \\
0110001 & 1011000 & 0000000 & 1100010 & 0001011 & 0101100 & 0101100 \\
1011000 & 0101100 & 0000000 & 0110001 & 1000101 & 0010110 & 0010110 \\
0101100 & 0010110 & 0000000 & 1011000 & 1100010 & 0001011 & 0001011 \\
0010110 & 0001011 & 0000000 & 0101100 & 0110001 & 1000101 & 1000101 \\
0001011 & 1000101 & 0000000 & 0010110 & 1011000 & 1100010 & 1100010 \\
\hline
1000101 & 1010100 & 0110100 & 0000000 & 0000111 & 1000110 & 1001001 \\
1100010 & 0101010 & 0011010 & 0000000 & 1000011 & 0100011 & 1100100 \\
0110001 & 0010101 & 0001101 & 0000000 & 1100001 & 1010001 & 0110010 \\
1011000 & 1001010 & 1000110 & 0000000 & 1110000 & 1101000 & 0011001 \\
0101100 & 0100101 & 0100011 & 0000000 & 0111000 & 0110100 & 1001100 \\
0010110 & 1010010 & 1010001 & 0000000 & 0011100 & 0011010 & 0100110 \\
0001011 & 0101001 & 1101000 & 0000000 & 0001110 & 0001101 & 0010011 \\
\hline
1001001 & 0100011 & 0001101 & 0111000 & 0000000 & 1001010 & 0101100 \\
1100100 & 1010001 & 1000110 & 0011100 & 0000000 & 0100101 & 0010110 \\
0110010 & 1101000 & 0100011 & 0001110 & 0000000 & 1010010 & 0001011 \\
0011001 & 0110100 & 1010001 & 0000111 & 0000000 & 0101001 & 1000101 \\
1001100 & 0011010 & 1101000 & 1000011 & 0000000 & 1010100 & 1100010 \\
0100110 & 0001101 & 0110100 & 1100001 & 0000000 & 0101010 & 0110001 \\
0010011 & 1000110 & 0011010 & 1110000 & 0000000 & 0010101 & 1011000 \\
\hline
1010001 & 0011001 & 0100011 & 1011000 & 1010100 & 0000000 & 1100001 \\
1101000 & 1001100 & 1010001 & 0101100 & 0101010 & 0000000 & 1110000 \\
0110100 & 0100110 & 1101000 & 0010110 & 0010101 & 0000000 & 0111000 \\
0011010 & 0010011 & 0110100 & 0001011 & 1001010 & 0000000 & 0011100 \\
0001101 & 1001001 & 0011010 & 1000101 & 0100101 & 0000000 & 0001110 \\
1000110 & 1100100 & 0001101 & 1100010 & 1010010 & 0000000 & 0000111 \\
0100011 & 0110010 & 1000110 & 0110001 & 0101001 & 0000000 & 1000011 \\
\hline
1001010 & 0010110 & 0100011 & 1100100 & 0001101 & 1100001 & 0000000 \\
0100101 & 0001011 & 1010001 & 0110010 & 1000110 & 1110000 & 0000000 \\
1010010 & 1000101 & 1101000 & 0011001 & 0100011 & 0111000 & 0000000 \\
0101001 & 1100010 & 0110100 & 1001100 & 1010001 & 0011100 & 0000000 \\
1010100 & 0110001 & 0011010 & 0100110 & 1101000 & 0001110 & 0000000 \\
0101010 & 1011000 & 0001101 & 0010011 & 0110100 & 0000111 & 0000000 \\
0010101 & 0101100 & 1000110 & 1001001 & 0011010 & 1000011 & 0000000
\end{array}\right)$$

autogp size = 1008

Latin square graph. NOT NEW.

$$
A_{10} = \left(\begin{array}{c|c|c|c|c|c|c}
0000000 & 1110000 & 1101000 & 1101000 & 1100100 & 1100010 & 1010100 \\
0000000 & 0111000 & 0110100 & 0110100 & 0110010 & 0110001 & 0101010 \\
0000000 & 0011100 & 0011010 & 0011010 & 0011001 & 1011000 & 0010101 \\
0000000 & 0001110 & 0001101 & 0001101 & 1001100 & 0101100 & 1001010 \\
0000000 & 0000111 & 1000110 & 1000110 & 0100110 & 0010110 & 0100101 \\
0000000 & 1000011 & 0100011 & 0100011 & 0010011 & 0001011 & 1010010 \\
0000000 & 1100001 & 1010001 & 1010001 & 1001001 & 1000101 & 0101001 \\
\hline
1000011 & 0000000 & 1010001 & 0110010 & 1000110 & 0101001 & 1000101 \\
1100001 & 0000000 & 1101000 & 0011001 & 0100011 & 1010100 & 1100010 \\
1110000 & 0000000 & 0110100 & 1001100 & 1010001 & 0101010 & 0110001 \\
0111000 & 0000000 & 0011010 & 0100110 & 1101000 & 0010101 & 1011000 \\
0011100 & 0000000 & 0001101 & 0010011 & 0110100 & 1001010 & 0101100 \\
0001110 & 0000000 & 1000110 & 1001001 & 0011010 & 0100101 & 0010110 \\
0000111 & 0000000 & 0100011 & 1100100 & 0001101 & 1010010 & 0001011 \\
\hline
1000101 & 1100010 & 0000000 & 0001011 & 0101100 & 0110001 & 0110001 \\
1100010 & 0110001 & 0000000 & 1000101 & 0010110 & 1011000 & 1011000 \\
0110001 & 1011000 & 0000000 & 1100010 & 0001011 & 0101100 & 0101100 \\
1011000 & 0101100 & 0000000 & 0110001 & 1000101 & 0010110 & 0010110 \\
0101100 & 0010110 & 0000000 & 1011000 & 1100010 & 0001011 & 0001011 \\
0010110 & 0001011 & 0000000 & 0101100 & 0110001 & 1000101 & 1000101 \\
0001011 & 1000101 & 0000000 & 0010110 & 1011000 & 1100010 & 1100010 \\
\hline
1000101 & 0010011 & 0110100 & 0000000 & 1001010 & 1000110 & 0011100 \\
1100010 & 1001001 & 0011010 & 0000000 & 0100101 & 0100011 & 0001110 \\
0110001 & 1100100 & 0001101 & 0000000 & 1010010 & 1010001 & 0000111 \\
1011000 & 0110010 & 1000110 & 0000000 & 0101001 & 1101000 & 1000011 \\
0101100 & 0011001 & 0100011 & 0000000 & 1010100 & 0110100 & 1100001 \\
0010110 & 1001100 & 1010001 & 0000000 & 0101010 & 0011010 & 1110000 \\
0001011 & 0100110 & 1101000 & 0000000 & 0010101 & 0001101 & 0111000 \\
\hline
1001001 & 1011000 & 0001101 & 1010100 & 0000000 & 0000111 & 1010001 \\
1100100 & 0101100 & 1000110 & 0101010 & 0000000 & 1000011 & 1101000 \\
0110010 & 0010110 & 0100011 & 0010101 & 0000000 & 1100001 & 0110100 \\
0011001 & 0001011 & 1010001 & 1001010 & 0000000 & 1110000 & 0011010 \\
1001100 & 1000101 & 1101000 & 0100101 & 0000000 & 0111000 & 0001101 \\
0100110 & 1100010 & 0110100 & 1010010 & 0000000 & 0011100 & 1000110 \\
0010011 & 0110001 & 0011010 & 0101001 & 0000000 & 0001110 & 0100011 \\
\hline
1010001 & 0100101 & 0100011 & 1011000 & 0111000 & 0000000 & 1001100 \\
1101000 & 1010010 & 1010001 & 0101100 & 0011100 & 0000000 & 0100110 \\
0110100 & 0101001 & 1101000 & 0010110 & 0001110 & 0000000 & 0010011 \\
0011010 & 1010100 & 0110100 & 0001011 & 0000111 & 0000000 & 1001001 \\
0001101 & 0101010 & 0011010 & 1000101 & 1000011 & 0000000 & 1100100 \\
1000110 & 0010101 & 0001101 & 1100010 & 1100001 & 0000000 & 0110010 \\
0100011 & 1001010 & 1000110 & 0110001 & 1110000 & 0000000 & 0011001 \\
\hline
1001010 & 1101000 & 0100011 & 0001110 & 1100010 & 1001100 & 0000000 \\
0100101 & 0110100 & 1010001 & 0000111 & 0110001 & 0100110 & 0000000 \\
1010010 & 0011010 & 1101000 & 1000011 & 1011000 & 0010011 & 0000000 \\
0101001 & 0001101 & 0110100 & 1100001 & 0101100 & 1001001 & 0000000 \\
1010100 & 1000110 & 0011010 & 1110000 & 0010110 & 1100100 & 0000000 \\
0101010 & 0100011 & 0001101 & 0111000 & 0001011 & 0110010 & 0000000 \\
0010101 & 1010001 & 1000110 & 0011100 & 1000101 & 0011001 & 0000000
\end{array}\right)
$$

autogp size = 126

NEW.

Index