NORTH-HOLLAND
MATHEMATICS STUDIES

114

ANNALS OF DISCRETE MATHEMATICS

26

Algorithms in Combinatorial Design Theory

C.J. COLBOURN
M.J. COLBOURN
editors

NORTH-HOLLAND

ALGORITHMS IN COMBINATORIAL DESIGN THEORY

NORTH-HOLLAND MATHEMATICS STUDIES 114
Annals of Discrete Mathematics (26)

NORTH-HOLLAND – AMSTERDAM • NEW YORK • OXFORD

ALGORITHMS IN COMBINATORIAL DESIGN THEORY

edited by

C. J. COLBOURN and M. J. COLBOURN
Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada

1985

NORTH-HOLLAND – AMSTERDAM • NEW YORK • OXFORD

ISBN: 0 444 87802 5

Publishers:
ELSEVIER SCIENCE PUBLISHERS B.V.
P.O. BOX 1991
1000 BZ AMSTERDAM
THE NETHERLANDS

Sole distributors for the U.S.A. and Canada:
ELSEVIER SCIENCE PUBLISHING COMPANY, INC.
52 VANDERBILT AVENUE
NEW YORK, N.Y. 10017
U.S.A.

Library of Congress Cataloging in Publication Data
Main entry under title:

Algorithms in combinatorial design theory.

(Annals of discrete mathematics ; 26) (North-Holland mathematics studies ; 114)
Includes bibliographies.
1. Combinatorial designs and configurations--Data processing. 2. Algorithms. I. Colbourn, C. J. (Charles J.), 1953- . II. Colbourn, M. J. (Marlene Jones), 1953- . III. Series. IV. Series: North-Holland mathematics studies ; 114.
QA166.25.A44 1984 511'.6 85-10371
ISBN 0-444-87802-5 (U.S.)

PRINTED IN THE NETHERLANDS

PREFACE

Recent years have seen an explosive growth in research in combinatorics and graph theory. One primary factor in this rapid development has been the advent of computers, and the parallel study of practical and efficient algorithms. This volume represents an attempt to sample current research in one branch of combinatorics, namely *combinatorial design theory*, which is algorithmic in nature.

Combinatorial design theory is that branch of combinatorics which is concerned with the construction and analysis of regular finite configurations such as projective planes, Hadamard matrices, block designs, and the like. Historically, design theory has borrowed tools from algebra, geometry and number theory to develop *direct* constructions of designs. These are typically supplemented by *recursive* constructions, which are in fact algorithms for constructing larger designs from some smaller ones. This lent an algorithmic flavour to the construction of designs, even before the advent of powerful computers.

Computers have had a definite and long-lasting impact on research in combinatorial design theory. Primarily, the speed of present day computers has enabled researchers to construct many designs whose discovery by hand would have been difficult if not impossible. A second important consequence has been the vastly improved capability for *analysis* of designs. This includes the detection of isomorphism, and hence gives us a vehicle for addressing enumeration questions. It also includes the determination of various properties of designs; examples include resolvability, colouring, decomposition, and subdesigns. Although in principle all such properties are computable by hand, research on designs with additional properties has burgeoned largely because of the availability of computational assistance.

Naturally, the computer alone is not a panacea. It is a well-known adage in design theory that computational assistance enables one to solve one higher order (only) than could be done by hand. This is a result of the "combinatorial explosion", the massive growth rate in the size of many combinatorial problems. Thus, research has turned to the development of practical algorithms which exploit computational assistance to its best advantage. This brings the substantial tools of computer science, particularly analysis of algorithms and computational complexity, to bear.

Current research on algorithms in combinatorial design theory is diverse. It spans the many areas of design theory, and involves computer science at every level from low-level implementation to abstract complexity theory. This volume is *not* an effort to survey the field exhaustively; rather it is an effort to present a collection of papers which involve designs and algorithms in an interesting way.

It is our intention to convey the firm conviction that combinatorial design theory and theoretical computer science have much to contribute to each other, and that there is a vast potential for continued research in the area. We would like to thank the contributors to the volume for helping us to illustrate the connections between the two disciplines. All of the papers were thoroughly refereed; we sincerely thank the referees, who are always the "unsung heroes and heroines" in a venture such as this. Finally, we would like especially to thank Alex Rosa, for helping in all stages from inception to publication.

Charles J. Colbourn and Marlene Jones Colbourn
Waterloo, Canada
March 1985

CONTENTS

Annals of Discrete Mathematics 26 (1985) 1–48

Computation of Some Parameters of Lie Geometries

A.E. Brouwer and A.M. Cohen

Centre for Mathematics and Computer Science
Kruislaan 413
1098 SJ Amsterdam
THE NETHERLANDS

Abstract

In this note we show how one may compute the parameters of a finite Lie geometry, and we give the results of such computations in the most interesting cases. We also prove a little lemma that is useful for showing that thick finite buildings do not have quotients which are (locally) Tits geometries of spherical type.

1. Introduction

The finite Lie geometries give rise to association schemes whose parameters are closely related to corresponding parameters of their associated Weyl groups. Though the parameters of the most common Lie geometries (such as projective spaces and polar spaces) are very well known, we have not come across a reference containing a listing of the corresponding parameters for geometries of Exceptional Lie type. Clearly, for the combinatorial study of these geometries the knowledge of these parameters is indispensible. The theorem in this paper provides a formula for those parameters of the association scheme that appear in the distance distribution diagram of the graph underlying the geometry. As a consequence of the theorem, we obtain a simple proof that the conditions in lemma 5 of [2] are fulfilled for the collinearity graph of any finite Lie geometry of type A_n, D_n, or E_m, $6 \le m \le 8$. (See remark 3 in section 4. The proof for the other spherical types, i.e. C_n, F_4, and G_2 is similar.) By means of the formula in the theorem, we have computed the parameters of the Lie geometries in the most interesting open cases for diagrams with single bonds only (A_n and D_n are well known, and are given as examples). The remaining cases follow similarly, but the complete listing of all parameters would take too much space.

2. Introduction to Geometries (following Tits [10])

A *geometry* over a set Δ (the set of *types)* is a triple $(\Gamma,*,t)$ where Γ is a set (the set of *objects* of the geometry), $*$ is a symmetric relation on Γ (the *incidence* relation) and t is a mapping (the *type* mapping) from Γ into Δ, such that for x, y $\in \Gamma$ we have ($t(x)=t(y)$ & $x*y$) if and only if $x=y$. (An example is provided by the collection Γ of all (nonempty proper) subspaces of a finite dimensional projective space, with t: $\Gamma \to \Delta = \mathbf{N}$, the rank function, and $*$ symmetrized inclusion (i.e., $x*y$ iff $x \subset y$ or $y \subset x$).)

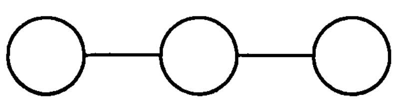
Often we shall refer to the geometry as Γ rather than as $(\Gamma,*,t)$.

A *flag* is a collection of pairwise incident objects. The *residue* Res(F) of a flag F is the set of all objects incident to each element of F. Together with the appropriate restrictions of $*$ and t, this set is again a geometry.

The *rank* of a geometry is the cardinality of the set of types Δ. The *corank* of a flag F is the cardinality of $\Delta \backslash t(F)$. A geometry is *connected* if and only if the (looped) graph $(\Gamma,*)$ is connected. A geometry is *residually connected* when for each flag of corank 1, Res(F) is nonempty, and for each flag of corank at least 2, Res(F) is nonempty and connected.

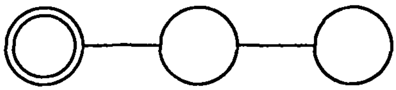
A *(Buekenhout-Tits) diagram* is a picture (graph) with a node for each element of Δ and with labelled edges. It describes in a compact way a set of axioms for a geometry Γ with set of types Δ as follows: whenever an edge $(d_1 d_2)$ is labelled with D, where D is a class of rank 2 geometries, then each residue of type $\{d_1,d_2\}$ of Γ must be a member of D. (Notice that a residue of type $\{d_1,d_2\}$ is the residue of a flag of type $\Delta \backslash \{d_1,d_2\}$.) In the following we need only two classes of rank 2 geometries. The first is the class of all projective planes, indicated in the diagram by a plain edge. The second is the class of all generalized digons, that is, geometries with objects of two types such that each object of one type is incident with every object of the other type. Generalized digons are indicated in the diagram by an invisible (i.e., absent) edge.

For example, the diagram

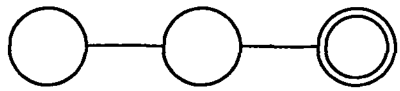


is an axiom system characterizing the geometry of points, lines, and planes of projective 3-space. Note that the residue of a line (i.e., the points on the line and the planes containing the line) is a generalized digon. Usually, one chooses one element of Δ and calls the objects of this type *points*. The residues of this type are called *lines*. Thus lines are geometries of rank 1, but all that matters is they constitute subsets of the point set. In the diagram the node corresponding to the points is encircled.

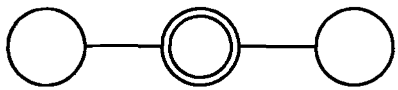
As an example, the principle of duality in projective 3-space asserts the isomorphism of the geometries



and

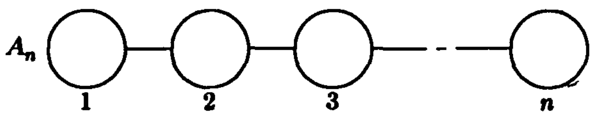


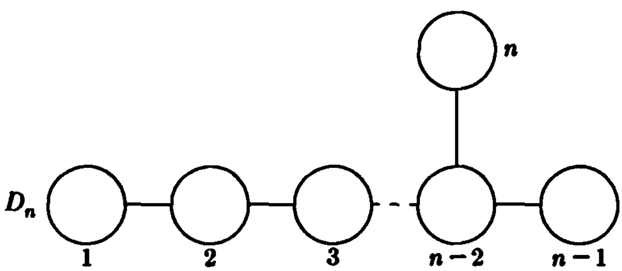
Grassmannians are geometries like

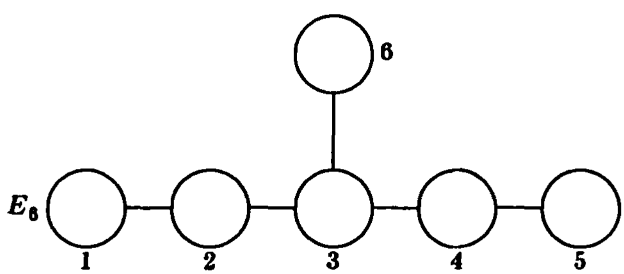


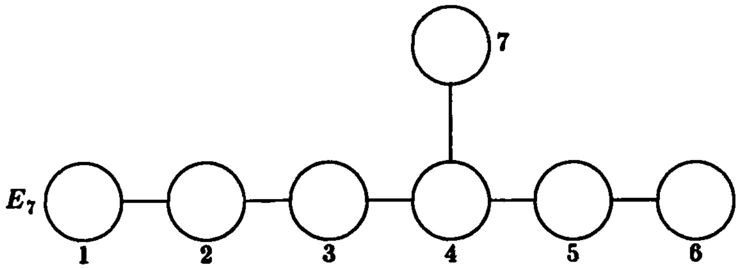
(Warning: points are objects of the geometry but lines are sets of points, and given a line, there need not be an object in the geometry incident with the same set of points.)

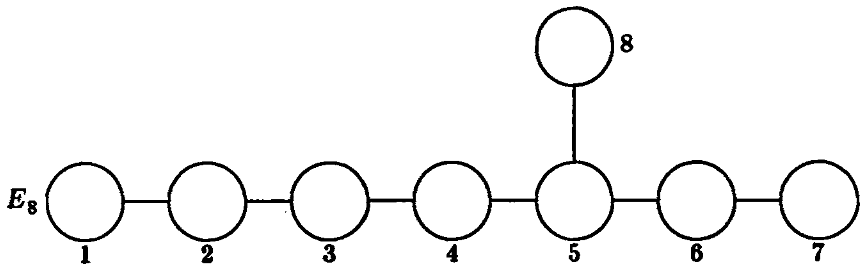
Let us write down some diagrams (with nodes labelled by the elements of Δ) for later reference.





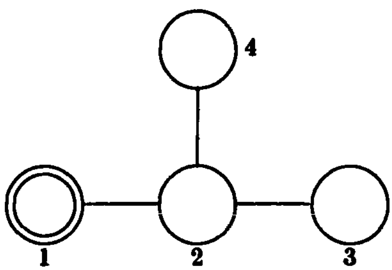






(Warning: in different papers different labellings of these diagrams are used.)

If one wants to indicate the type corresponding to the points, it is added as a subscript. For example, $D_{4,1}$ denotes a geometry belonging to the diagram



It is possible to prove that if Γ is a finite residually connected geometry of rank at least 3 belonging to one of these diagrams having at least three points on each line then the number of points on each line is $q+1$ for some prime power q, and given a prime power q there is a unique geometry with given diagram and $q+1$ points on each line. We write $X_n(q)$ for this unique geometry, where X_n is the name of the diagram (cf. Tits [9] Chapter 6, and [2]).

[For example, $A_n(q)$ is the geometry of the proper nonempty subspaces of the projective space PG(n,q). Similarly, $D_n(q)$ is the geometry of the nonempty totally isotropic subspaces in PG($2n-1$,q) supplied with a nondegenerate quadratic form of maximal Witt index. Finally, $D_{n,1}(q)$ is an example of a polar space.]

A remark on notation: ':=' means "is by definition equal to" or "is defined as".

3. Distance Distribution Diagrams for Association Schemes

An *association scheme* is a pair $(X,\{R_0,\ldots,R_s\})$ where X is a set and the R_i $(0 \leq i \leq s)$ are relations on X such that $\{R_0,\ldots,R_s\}$ is a partition of $X \times X$ satisfying the following requirements:

(i) $R_0 = I$, the identity relation.

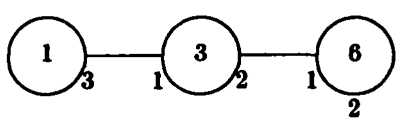
(ii) for all i, there exists an i' such that $R_i^T = R_{i'}$

(iii) Given $x,y \in X$ with $(x,y) \in R_i$, then the number $p_{jk}^i = |\{z\colon (x,z) \in R_j$ and $(y,z) \in R_k\}|$ does not depend on x and y but only on i.

Usually we shall write v for the total number of points of the associated scheme, i.e. $v = |X|$. The obvious example of an association scheme is the situation where a group G acts transitively on a set X. In this case one takes for $\{R_0,\ldots,R_s\}$ the partition of $X \times X$ into G-orbits, and requirements (i)-(iii) are easily verified.

Assume that we have an association scheme with a fixed symmetric nonidentity relation R_1 (i.e., $R_1^T = R_1$). Clearly (X,R_1) is a graph. Now one may draw a diagram displaying the parameters of this graph by drawing a circle for each relation R_i, writing the number $k_i = |\{z\colon (x,z) \in R_i\}| = p_{ii'}^0$ where $x \in X$ is arbitrary inside the circle, and joining the circles for R_i and R_j by a line carrying the number p_{j1}^i at the (R_i)-end whenever $p_{j1}^i \neq 0$. (Note that $k_i p_{j1}^i = k_j p_{i1}^j$ so that p_{j1}^i is nonzero iff p_{i1}^j is nonzero.) When $i = j$, one usually omits the line and just writes the number p_{i1}^i next to the circle for R_i.

For example, the Petersen graph becomes a symmetric association scheme, i.e., one for which $R_i^T = R_i$ for all i when we define $(x,y) \in R_i$ iff $\mathrm{d}(x,y) = i$ for $i = 0,1,2$. We find the diagram



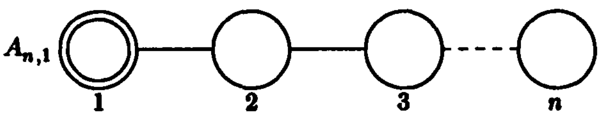
More generally, a graph is called *distance regular* when $(x,y) \in R_i$ iff $\mathrm{d}(x,y) = i$ $(0 \leq i \leq diam(G))$ defines an association scheme.

When (X,R_1) is a distance regular graph, or, more generally, when the matrices I, A, A^2, ..., A^s are linearly independent (where A is the 0-1 matrix of R_1, i.e., the adjacency matrix of the graph), then the p_{j1}^i suffice to determine all p_{jk}^i. On the other hand, when the association scheme is not symmetric but R_1 is, then clearly not all R_j can be expressed in terms of R_1.

In this note our aim is to compute the parameters p^i_{jk} for the Lie geometries $X_{m,n}(q)$ where X_m is a (spherical) diagram with designated 'point'-type n, and the association scheme structure is given by the group of (type preserving) automorphisms of $X_{m,n}(q)$ - essentially a Chevalley group. In the next section we shall give formulas valid for all Chevalley groups and in the appendix we list results in some of the more interesting cases. Let us do some examples explicitly. (References to words in the Weyl group will be explained in the next section.)

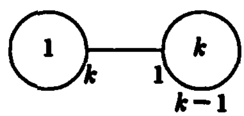
Usually we give only the p^i_{j1}; the general case follows in a similar way.

Example 1.

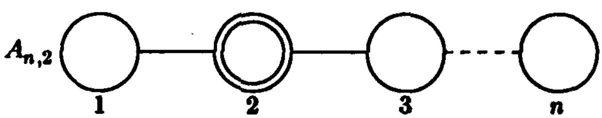


The collinearity graph of points in a projective space is a clique: any two points are adjacent (collinear). Thus our diagram becomes

$$v=\frac{q^{n+1}-1}{q-1},\ k=\frac{q^n-1}{q-1}q=v-1.$$



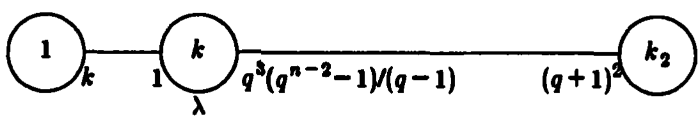
Example 2.



Now we have the graph of the projective lines in a projective space, two projective lines being adjacent whenever they are in a common plane (and have a projective point in common).

[N.B.: the lines of this geometry are pencils of $q+1$ projective lines in a common plane and on a common projective point.]

Our diagram becomes



Weyl words: "" "2" "2312"

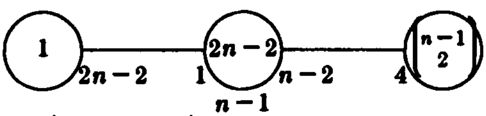
$$v=\frac{(q^{n+1}-1)(q^n-1)}{(q^2-1)(q-1)}$$

$$k=q(q+1)\frac{q^{n-1}-1}{q-1}$$

$$\lambda=q-1+q^2+q^2\frac{q^{n-2}-1}{q-1}$$

$$k_2=\frac{q^{n-1}-1}{q^2-1}\frac{q^{n-2}-1}{q-1}q^4$$

For $q=1$ (the 'thin' case) this becomes the diagram for the triangular graph:

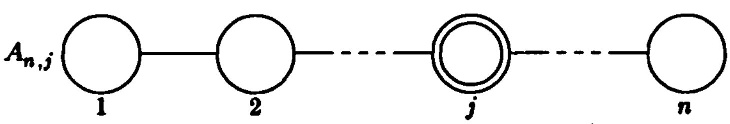


$v=\binom{n+1}{2}$

[Clearly $\lambda_i:=p^i_{1i}=k-\sum_{j\neq i}p^i_{1j}$. Often, when λ_i does not have a particularly nice form, we omit this redundant information.]

Notice how easily the expressions for v,k,k_2,λ can be read off from the Buekenhout-Tits diagram: for example, $\lambda=\lambda(x,y)$ first counts the $q-1$ points on the line xy, then the remaining q^2 points of the unique plane of type $\{1,2\}$ containing this line and finally the remaining q^2 points of the planes of type $\{2,3\}$ containing this line.

Example 3.



This is the graph of the j-flats (subspaces of dimension j) in projective n-space, two j-flats being adjacent whenever they are in a common $(j+1)$-flat (and have a $(j-1)$-flat in common). The graph is distance regular with diameter $min(j,n+1-j)$. Parameters are

$$v=\frac{(q^{n+1}-1)(q^n-1)\ldots(q^{n+2-j}-1)}{(q^j-1)(q^{j-1}-1)\ldots(q-1)}=:\begin{bmatrix}n+1\\ j\end{bmatrix}_q$$

$$k=q^{i^2}\begin{bmatrix} j \\ i \end{bmatrix}_q \begin{bmatrix} n-j+1 \\ i \end{bmatrix}_q$$

$$b_i := p^i_{1,i+1} = q^{2i+1}\begin{bmatrix} j-i \\ 1 \end{bmatrix}_q \begin{bmatrix} n-j-i+1 \\ 1 \end{bmatrix}_q$$

$$c_i := p^i_{1,i-1} = \begin{bmatrix} i \\ 1 \end{bmatrix}_q^2$$

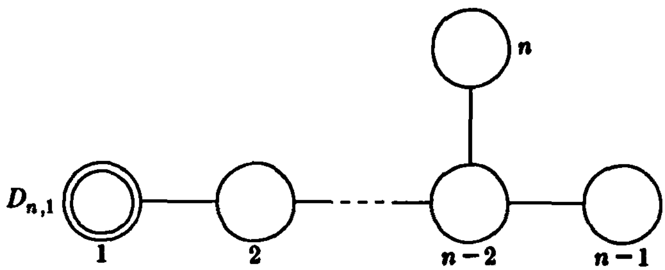
The parameters for the thin case have $q=1$ and binomial instead of Gaussian coefficients; we find the Johnson scheme $\binom{n+1}{j}$.

The Weyl words (minimal double coset representatives in the Weyl group) have the following shape: for double coset i in $A_{n,j}$ the representative is

$$w_i := \text{“}j,j+1,\ldots,j+i-1,j-1,j,\ldots,j+i-2,\ldots\ldots,j-i+1,j-i+2,\ldots,j\text{”}.$$

Note that w_i has length i^2, the power of q occurring in k_i.

Example 4.

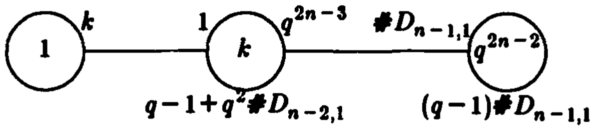


($n \geq 3$; $D_{2,1}$ is the direct product $A_{1,1} \times A_{1,1}$, i.e., a $(q+1)\times(q+1)$ grid.)

$$v = \# D_{n,1} = \frac{(q^n-1)(q^{n-1}+1)}{q-1}$$

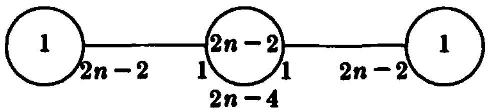
$$k = q\# D_{n-1,1} = q\frac{(q^{n-1}-1)(q^{n-2}+1)}{q-1}$$

Diagram:



Thin case:

$$v=2n, k=2n-2$$



This is K_{2n} minus a complete matching.

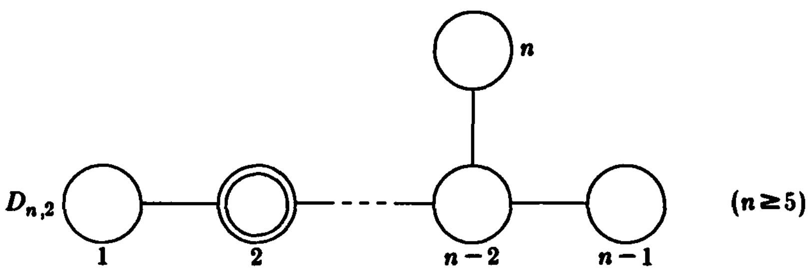
The Weyl words are:

"" for double coset 0,

"1" for double coset 1, and

"1 2 3 $\cdots$ $n-3$ $n-2$ n $n-1$ $n-2$ $\cdots$ 1" for double coset 2.

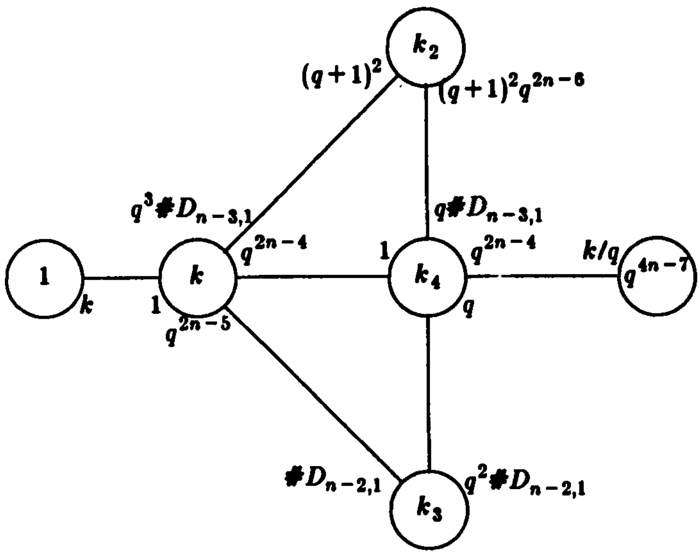
Example 5.



$$v=\frac{\#D_{n,1}\#D_{n-1,1}}{\#A_{1,1}}=\frac{(q^n-1)(q^{n-1}+1)(q^{n-1}-1)(q^{n-2}+1)}{(q^2-1)(q-1)}$$

$$k=q\#A_{1,1}\#D_{n-2,1}=q(q+1)\frac{(q^{n-2}-1)(q^{n-3}+1)}{q-1}$$

Diagram (for $n>4$):



Double coset 1 contains adjacent points, i.e., lines of the polar space in a common plane. Shortest path in the geometry: 2-3-2 (unique).

Double coset 2 contains the points at 'polar' distance two, belonging to the Weyl word "2312", i.e., in a polar space $A_{3,2}$. (I.e., lines of the polar space in a common t.i. subspace). Thus

$$k_2=\#D_{n-2,2}k_2(A_{3,2})=\frac{q^{2n-6}-1}{q^2-1}\frac{q^{n-2}-1}{q-1}(q^{n-4}+1)q^4$$

Shortest path in the geometry: 2-4-2 (unique). Double coset 3 contains points incident with a common 1-object, so that the Weyl word is the one for double coset 2 in $D_{n-1,1}$ (relabelled):

$$\text{“}2\ 3\ \cdots\ n-3\ n-2\ n\ n-1\ n-2\ \cdots\ 2\text{”}.$$

(These are intersecting lines not in a common t.i. plane.) Thus

$$k_3=\#A_{1,1}k_2(D_{n-1,1})=(q+1)q^{2n-4}.$$

Shortest path in the geometry: 2-1-2 (unique).

Double coset 4 contains points with shortest path 2-1-3-2 (unique); the Weyl word is

$$\text{“}2\ 3\ \cdots\ n-3\ n-2\ n\ n-1\ n-2\ \cdots\ 3\ 1\ 2\text{”},$$

the reduced form of the product of the word we found for double coset 3 and the word "212" describing adjacency in $A_{2,2}$. Thus

$$k_4 = \#D_{n-2,1}q^2(\#D_{n-1,1}-(q+1)-q^2\#D_{n-3,1}) = \frac{q^{n-2}-1}{q-1}(q^{n-3}+1)(q+1)q^{2n-3}$$

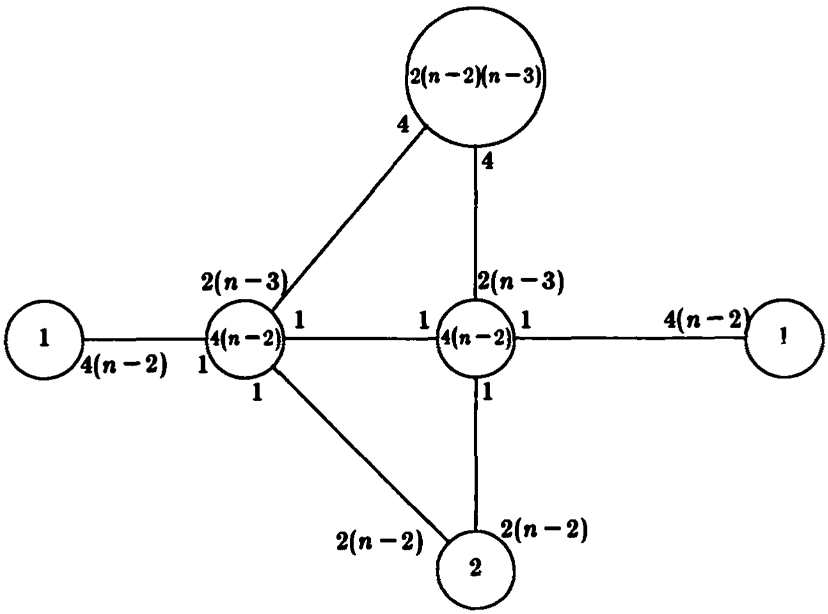
Double coset 5 contains the remaining q^{4n-7} points (the lines of the polar space in general position). Shortest path in the geometry: 2-1-2-1-2 (not unique). The Weyl word is

$$\text{“}2\ 3\ \cdots\ n-1\ 1\ 2\ 3\ \cdots\ n-2\ n\ n-2\ \cdots\ 2\ 1\ n-1\ \cdots\ 3\ 2\text{”}$$

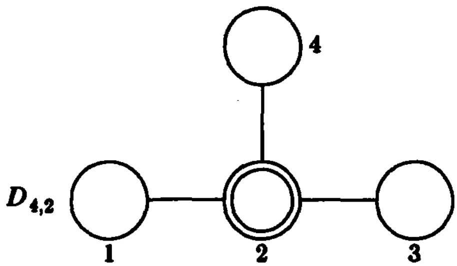
of length $4n-7$.

The thin case is:

$$v = 2n(n-1), \quad k = 4(n-2)$$



Example 6.



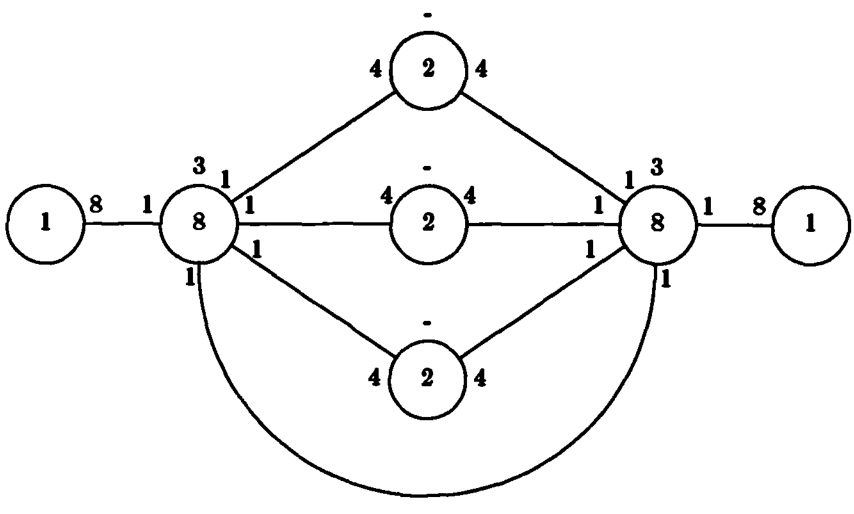
As before we find

$$v = \frac{q^6-1}{q^2-1}\,\frac{q^4-1}{q-1}(q^2+1) = \frac{q^6-1}{q-1}(q^2+1)^2$$

and $k = q(q+1)^3$.

This time the thin diagram is

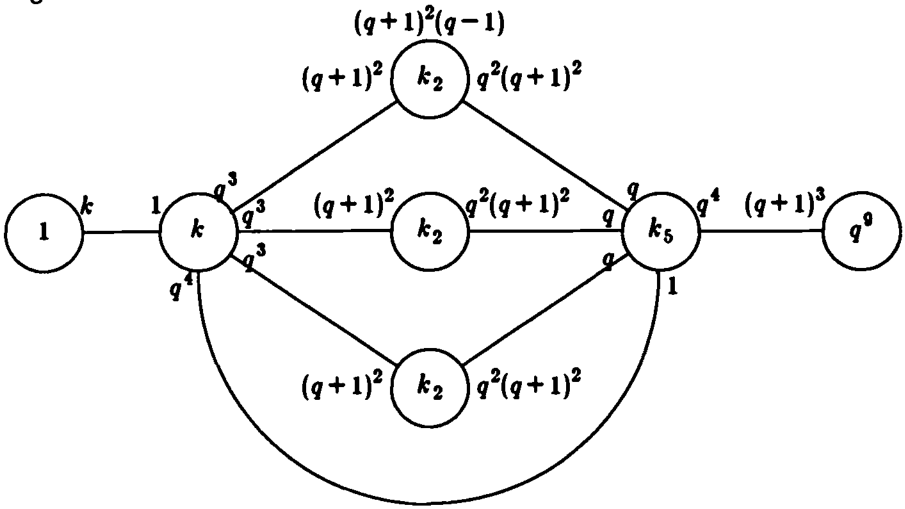
$$v=24, \quad k=8$$



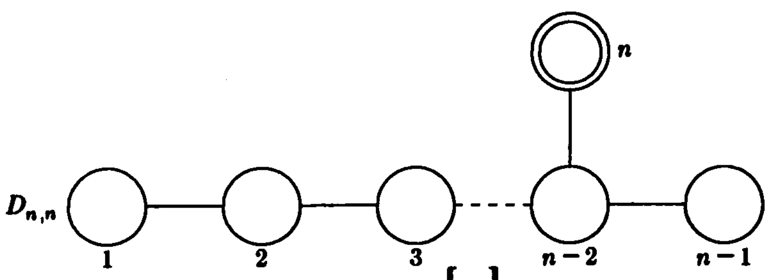
and we see that the number of classes is one higher than before. This is caused by the fact that we can distinguish here between shortest paths 2-4-2 and 2-3-2, while in the general case ($n \geq 5$) both $2-n-2$ and $2-(n-1)-2$ are equivalent to $2-3-2$. Thus, our previous double coset 2 splits here into two halves.

Double coset	Weyl word	Cardinality	Shortest path (unique)
0	“”	1	2
1	“2”	$q(q+1)^3$	2-{1,3,4}-2
2	“2312”	$q^4(q+1)$	2-4-2
3	“2412”	$q^4(q+1)$	2-3-2
4	“2432”	$q^4(q+1)$	2-1-2
5	“24312”	$q^5(q+1)^3$	2-1-{3,4}-2
6	“231242132”	q^9	

Diagram:



Example 7.



This graph is distance regular of diameter $\left[\frac{n}{2}\right]$.

We have

$$v=(q^{n-1}+1)(q^{n-2}+1)\ldots(q+1)$$

$$k=q\begin{bmatrix}n\\2\end{bmatrix}_q$$

$$k_i=q^{\binom{2i}{2}}\begin{bmatrix}n\\2i\end{bmatrix}_q$$

$$b_i=q^{4i+1}\begin{bmatrix}n-2i\\2\end{bmatrix}_q$$

$$c_i=\begin{bmatrix}2i\\2\end{bmatrix}_q$$

Note that when $n=2m$, then $k_m=q^{m(2m-1)}$. Also, note that in the case $n=4$ these parameters reduce to those we found for $D_{4,1}$.

Two points have distance $\leq i$ (for $0\leq i\leq n$) iff there is a path $n-(n-2i)-n$ in the geometry. When n is even, then two points at distance $\frac{n}{2}$ ("in general position") are not incident to a common object. (Note that $k=\#A_{n-1,2}q$ and, more generally, that

$$k_i=\#A_{n-1,2i}\,k_i(D_{2i,2i})=q^{i(2i-1)}\#A_{n-1,2i}.$$

The values for b_i and c_i follow similarly. The value for v follows by induction, and when $n=2m$ then k_m is found from $k_m=v-\sum_{i<m}k_i$.)

The Weyl word corresponding to distance i is the same one (after relabelling) as in $D_{2i,2i}$, namely:

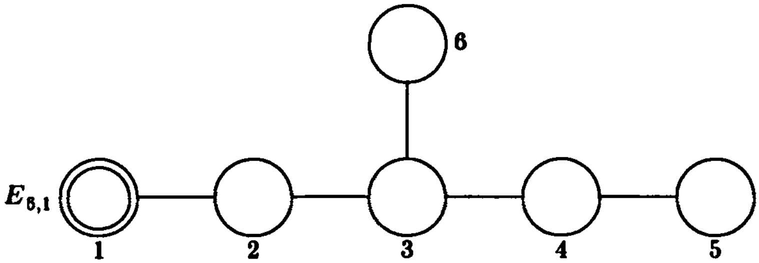
$$\text{“}n\ \ n-2\ \ n-1\ \ n-3\ \ n-2\ \ n\ \ n-4\ \ n-3\ \ n-2\ \ n-1\ \ \cdots\text{”}$$

of length

$$1+2+3+4+\cdots+2i-1 = i(2i-1).$$

In the thin case we have $v = 2^{n-1}$, $k=\binom{n}{2}$, and the graph is that of the binary vectors of even weight and length n where the distance is the Johnson distance, i.e., half the Hamming distance.

Example 8 (see Tits [8]).



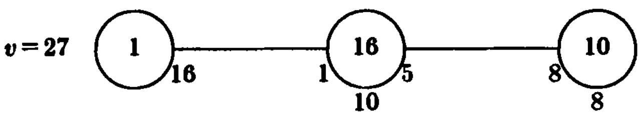
This graph is strongly regular (i.e., distance regular with diameter 2). We have

$$v=\frac{q^{12}-1}{q^4-1}\,\frac{q^9-1}{q-1}$$

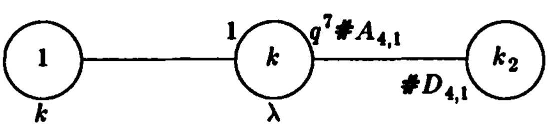
and

$$k=q\#D_{5,5}=q\frac{q^8-1}{q-1}(q^3+1).$$

The thin case gives diagram



the Schläfli graph; this is the complement of the collinearity graph of the generalized quadrangle GQ(2,4). In general we find the diagram

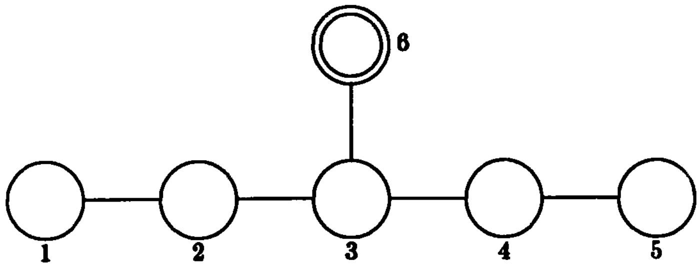


where $k_2=q^8\#D_{5,1}$ and $\lambda=q-1+q^2\#A_{4,2}$.

Double coset 1 corresponds to the shortest path $1-2-1$ and has Weyl word "1". Double coset 2 corresponds to the shortest path $1-5-1$ and has Weyl word "12364321", as in $D_{5,1}$.

Example 9.

$E_{6,6}$



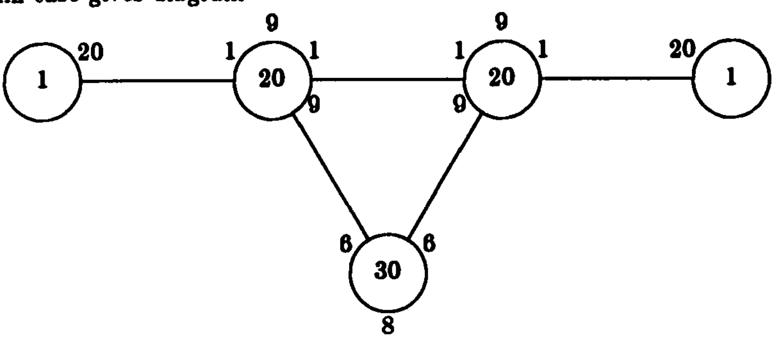
This graph has

$$v=\frac{q^9-1}{q-1}(q^6+1)(q^4+1)(q^3+1)$$

and

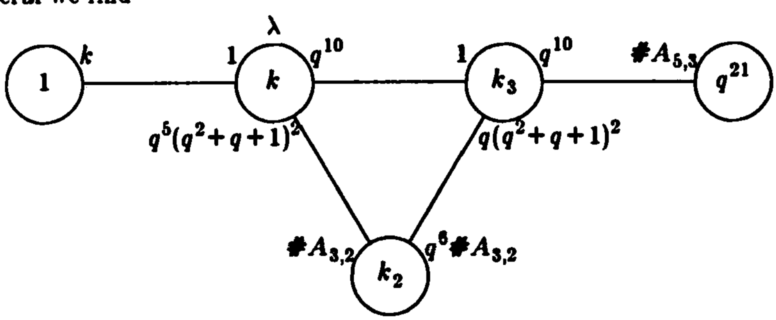
$$k=q\#A_{5,3}=q(q^2+1)(q^3+1)\frac{q^5-1}{q-1}$$

The thin case gives diagram



with $v\ =\ 72$.

In general we find



with $k_2=\#A_{5,1}\#A_{4,1}q^6$ and $k_3=q^{10}k$ and $\lambda=q-1+q^2(q^2+q+1)^2$. Double coset 1 corresponds to shortest path 6-3-6 and has Weyl word "6". Double coset 2 corresponds to shortest path 6-{1,5}-6 and has Weyl word "634236" (of $D_{4,1}$).

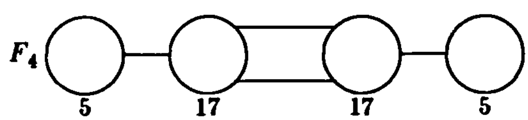
Double coset 3 corresponds to shortest path 6-1-4-6 (or, equivalently, 6-5-2-6) and has Weyl word "6345 234 1236". Double coset 4 has Weyl word "6345 234 1236345 234 1236".

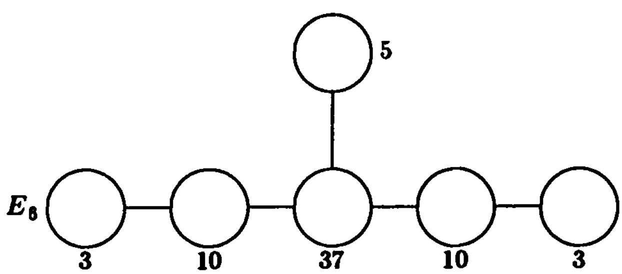
Example 10.

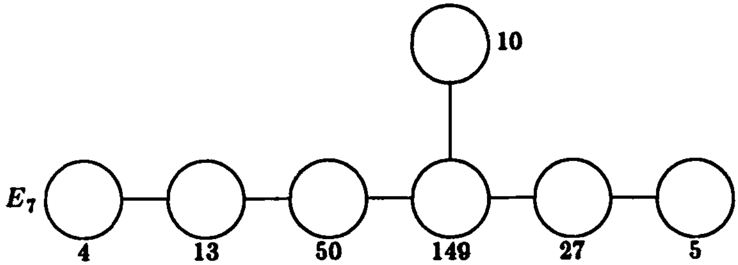
The case of type $F_{4,1}$ has been treated in Cohen [6].

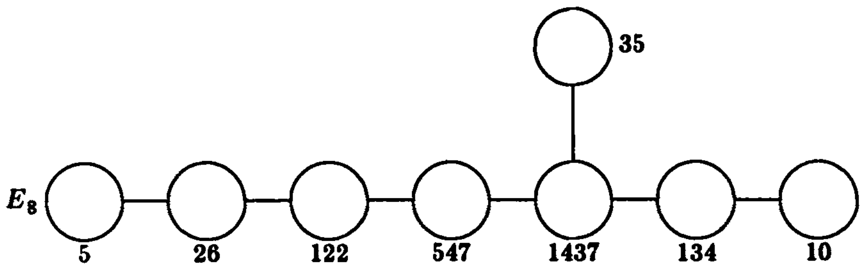
Up to now all our computations were easy and straightforward, mainly because of the limited permutation ranks (number of classes of these association schemes) and the fact that $A_{n,1}$, $D_{n,1}$, and $E_{6,1}$ have diameter at most two. Continuing in this vein we quickly encounter difficulties. $E_{7,1}$ is still distance regular with diameter three and $E_{7,6}$ and $E_{8,1}$ have diagrams like $E_{6,6}$ (and these three cases are easily done by hand) but for instance $E_{7,4}$ has 149 classes (double cosets) and all geometric intuition is lost; in the next section we describe how parameters for these Lie geometries can be mechanically derived by means of some computations in the Weyl group. In a way, this means that it suffices to consider the case $q=1$. Now everything is finite and a computer can do the work.

In the appendix we give computer output describing $E_{7,1}$, $E_{7,6}$, $E_{7,7}$, $E_{8,1}$, $E_{8,7}$, and $E_{8,8}$, in other words, the geometries belonging to the 'end nodes' of the diagrams E_7 and E_8. For E_7 we also computed the parameters on the remaining nodes, but listing these would take too much room. We therefore content ourselves with the presentation of the permutation ranks for the Chevalley groups of type F_4, E_n ($6 \leq n \leq 8$); to each node r in the diagram below is attached the permutation rank of the Chevalley group of the relevant type on the maximal parabolic corresponding to r.









4. Reduction to the Weyl group

In this section, G is a Chevalley group $X_n(q)$ of type X_n over a finite field $\mathbf{F}_q$. We shall rely heavily on Carter [4], to which the reader is referred for details. Though with a little more care, all statements can be adapted so that they are also valid for twisted Chevalley groups, for the sake of simplicity, we shall only consider the case of an untwisted Chevalley group G. To G we can associate a split saturated Tits system (B,N,W,R), cf. Bourbaki [1], consisting of subgroups B,N of G such that G is generated by them, and of a Coxeter system (W,R) with the following properties:

(i) $H = B \cap N$ is a normal subgroup of N and $W = N/H$.

(ii) For any $w \in W$ and $r \in R$,

(ii)' $BwBrB \subset BwB \cup BwrB$

(ii)'' ${}^rB \subset B$

(iii) (split) There is a normal subgroup U of B with $B = UH$ and $U \cap H = \{1\}$.

(iv) (saturated) $\bigcap_{w \in W} {}^wB = H$.

Here and below, ${}^wA$ stands for wAw^{-1} if A is a subset of G invariant under conjugation by H. Notice that ${}^wB$ and Bw are well defined. We shall briefly recall how a Tits system may be obtained. Start with a Coxeter system (W,R) where W is a Weyl group of type X_n. Let Φ be a root system for W. A set of mutually obtuse roots corresponding to the subset R (of *fundamental reflections)* forms a set of *fundamental roots*. Now any root $\alpha\epsilon\Phi$ is an integral linear combination of the fundamental roots such that either all coefficients are nonnegative or all coefficients are nonpositive. In the former case, α is called *positive*, denoted $\alpha>0$; in the latter case, α is called *negative*, denoted $\alpha<0$.

Now choose a Cartan subgroup H in G, and denote by X_α for $\alpha\epsilon\Phi$ the root subgroup with respect to α (viewed as a linear character of H). Thus H normalizes each X_α. Next, let N be the normalizer of H in G. Then $W=N/H$ permutes the X_α ($\alpha\epsilon\Phi$) according to ${}^wX_\alpha = X_{w\alpha}$ ($w\epsilon W$).

Now $U=\prod_{\alpha>0} X_\alpha$ is a subgroup of G normalized by H, so that $B=UH$ is a subgroup of G with $B\cap N=H$. This explains how B,N,W,R,U occur in G. We need some more subgroups of G. Given $w\epsilon W$, set

$$U_w^- := \prod_{\alpha>0,w^{-1}\alpha<0} X_\alpha.$$

It is of crucial importance to the computations below that

$$|U_w^-| = q^{l(w)}$$

for every $w\epsilon W$, where $l(w)$ denotes the length of w with respect to R. (For a proof, see Carter [4] 8.6; notice that our definition of U_w^- differs from Carter's in that our U_w^- coincides with his $U_{w^{-1}}^-$.) Observe that U_w^- is a subgroup of U, for if we let w_0 denote the unique longest element in W with respect to l, then w_0 is an involution satisfying $U_w^- = U \cap {}^{ww_0}U$ (and also $U \cap {}^{w_0}U = \{1\}$). Fix $r\epsilon R$ and write $J = R\backslash\{r\}$, $W_J = \langle J\rangle$, the subgroup of W generated by J, and $P=BW_JB$. Then P is a socalled maximal parabolic subgroup of G (associated with r). We are interested in the graph $\Gamma=\Gamma(G,P)$ defined as follows. Its vertices are the cosets xP in G (for $x\epsilon G$), two vertices xP,yP being adjacent when $y^{-1}x\epsilon PrP$.

In this graph, xP and yP have distance $d(xP,yP)\leq e$ if and only if $y^{-1}x\epsilon P\langle r\rangle \cdots \langle r\rangle P$ (a product of $2e+1$ terms). Let us first compute the number v of vertices of this graph.

Lemma 1. Each coset xP has a unique representation $xP=uwP$ where $u\epsilon U_w^-$ and w is a right J-reduced element of W, i.e.,

$$w\epsilon L_J := \{w\epsilon W \mid l(ww')\geq l(w) \text{ for } all\ w'\epsilon W_J\}.$$

Proof:

xB has a (unique) representation $xB=uwB$ with $w\epsilon W$, $u\epsilon U_w^-$ (see Carter [4], Theorem 8.4.3). Thus $xP=uwP$ and obviously we may take $w\epsilon L_J$ (cf. Bourbaki [1], Chap. IV, §1 Exercice 3). Suppose $uwP=u'w'P$. Then $w'\epsilon BwBW_JB$ so that $w' = ww''$ with $w'' \in W_J$, but since $w,w'\epsilon L_J$ it follows that

$w'=w$. We assert that $P\cap w^{-1}Bw \subset B$. (See [5], Proposition p. 63; since this reference is not easily accessible we repeat the argument.) Let $w=r_1r_2\cdots r_t$ be an expression of w as a product of $t=l(w)$ reflections in R. Denote by S the set of elements of the form $r_{i_1}r_{i_2}\cdots r_{i_s}$ with $i_1<i_2<\cdots<i_s$. Then $W_J\cap S^{-1}w = \{1\}$ since $wW_J\cap S = \{w\}$ (w is the only element in S with length at least $l(w)$). Hence, $P\cap w^{-1}Bw \subset BW_JB\cap Bw^{-1}BwB \subset BW_JB\cap BS^{-1}wB = B(W_J\cap S^{-1}w)B = B$, as asserted. Now $u^{-1}u' \in wPw^{-1}\cap U_w^- = w(P\cap w^{-1}Uw\cap w_0Uw_0^{-1})w^{-1} \subset w(B\cap w_0Uw_0^{-1})w^{-1} = \{ww^{-1}\} = \{1\}$ since $B\cap {}^{w_0}U = 1$ (see Carter [4], Lemma 7.1.2). Thus $u=u'$. ●

Proposition 1. The graph $\Gamma(G,P)$ has v vertices, where

$$v=\sum_{w\in L_J} q^{l(w)}.$$

Proof:

A straightforward consequence of the formula $|U_w^-| = q^{l(w)}$ for $w\in W$ and lemma 1. ●

Remark 1. Of course, we also have the multiplicative formula

$$v=|G/P|=\prod_{i=1}^{n}\frac{q^{d_i}-1}{q^{e_i}-1}$$

where $d_1, ..., d_n$ are the degrees of the Weyl group W, $e_2, ..., e_n$ are the degrees of the Weyl group W_J and $e_1 = 1$ (cf. Carter [4]).

Next, we want to put the structure of an association scheme on this graph. The group G acts by left multiplication on the cosets xP, and clearly this action is transitive. Thus we find an association scheme. The collections of cosets in a fixed relation with a given coset, say P, are the double cosets PxP. The pair (xP,yP) has relation $G(xP,yP)$, labelled with $Px^{-1}yP$. We see that a relation PxP is symmetric iff $PxP = Px^{-1}P$, and this holds in particular for $x=r$.

Lemma 2. Each double coset PxP has a unique representation $PxP=PwP$ where w is an element of W that is both left and right J-reduced, i.e.,

$$w\in D_J:=\{w\in W \mid w \textit{ is the unique shortest word of } W_JwW_J\}.$$

Proof:

See Bourbaki [1] Chap. IV §1 Exercice 3. ●

Proposition 2. The association scheme $\Gamma(G,P)$ has valencies k_i (belonging to the relation PiP) for $i\in D_J$, where

$$k_i=\sum_{w\in L_J\cap W_Ji} q^{l(w)}.$$

Proof:

Obvious. ●

Remark 2. If $i \epsilon D_J$, then $iW_J i^{-1} \cap W_J = W_{iJi^{-1} \cap J}$ by Solomon [7], so substitution of $q=1$ in the above formula for k_i leads to the equation $|L_J \cap W_J i| = \frac{|W_J|}{|W_{iJi^{-1}\cap J}|}$.

Finally, we come to the parameters p^i_{jk}. It is more convenient to label the relations (such as i,j,k) by elements from D_J than by $0,1,\ldots,s$ as in Section 2. Therefore, we shall use these new labels; 1 now stands for the "old 0", and r for adjacency, i.e., the "old 1". We shall confine ourselves to giving p^i_{jr}.

Theorem. Let $i,j \epsilon D_J$. Then the number of points (i.e., cosets) in $iPrP \cap PjP$ is

$$p^i_{jr} = \sum_{w \epsilon L \cap A\ ,\ l(iw) > l(iwr)} q^{l(w)} + \sum_{w \epsilon L \cap A\ ,\ l(iw) < l(iwr)} q^{l(wr)} + \sum_{w \epsilon L \cap Ar\ ,\ l(iw) < l(iwr)} q^{l(wr)}(q-1)$$

where $L := L_J \cap W_J r$ and $A := i^{-1} W_J j W_J$.

Proof:

Clearly,

$$W_J j W_J = \dot{\bigcup_{w \epsilon L}} wW_J$$

Consequently,

$$iPrP = iBW_J BrBW_J B = iBW_J rW_J B = \dot{\bigcup_{w \epsilon L}} iBwP$$

Now we want to write each set $iBwP$ as a union of cosets uwP as in lemma 1. For $g \epsilon G$ and K a subgroup of G define ${}^gK := gKg^{-1}$ and $K^\# = K \backslash \{1\}$. It is well known that for any $u \epsilon W$ we have *if* $l(iu) = l(i) + l(u)$ *then* ${}^i(U_u^-) \subseteq U_{iu}^-$. (See Cohen [5] Lemma 2.11.) Notice that $w = vr$ for some $v \epsilon W_J$ with $l(iv) = l(i) + l(v)$ and $l(vr) = l(v) + 1$.

Distinguish two cases:

If $l(iw) > l(iv)$ then

$$iBwB = iU_w^- wB = {}^i(U_w^-) iwB$$

and we have ${}^i(U_w^-) \subseteq U_{iw}^-$ as desired.

If $l(iw) < l(iv)$ then

$$iBwB = iBvBrB = {}^{i}(U_v^-)ivBrB = {}^{i}(U_v^-)\cdot(iwB \mathbin{\dot{\cup}} {}^{iw}((U_r^-)^{\#})ivB)$$

and we have ${}^{i}(U_v^-) \subseteq U_{iw}^-$, ${}^{i}(U_v^-)\cdot{}^{iw}(U_r^-) \subset U_{iv}^-$ as desired. (For the inclusion ${}^{i}(U_v^-) \subseteq U_{iw}^-$ note that v cannot change the sign of the root corresponding to r since $v \in W_J$.)

Now in order to count how many of the cosets uwP fall into a given double coset PjP we need only observe that $uwP \subseteq PjP$ iff $w \in W_J j W_J$, and that distinct $w \in L$ lead to distinct cosets iwP. ●

Corollary. Given two vertices x_1P, x_2P of Γ at mutual distance d, the number of vertices at distance $d-1$ to x_1P and adjacent to x_2P is congruent to 1 (mod q), and the number of vertices at distance d to x_1P and adjacent to x_2P is congruent to -1 (mod q). Also, the valency k is congruent to 0 (mod q).

Proof:

From "$w \in W_J r$ iff $l(w) \geq 1$" and the expression given for $k = k_r$ we see that $k \equiv 0$ (mod q). Next, from the previous theorem we obtain that

$$p_{jr}^i \equiv \delta(ir \in W_J j W_J) + (q-1)\cdot\delta(i \in W_J j W_J) \pmod q$$

where $\delta(T)$ for a predicate T denotes 1 if T is true and 0 otherwise. Thus, all p_{jr}^i are congruent to 0 (mod q) except p_{ir}^i which is congruent to -1 (mod q) and $p_{\bar{i}r}^i$ which is congruent to 1 (mod q) -- where $\bar{i}$ is defined by $ir \in W_J \bar{i} W_J$. Clearly $d(P,\bar{i}P) = d(P,iP) - 1$. ●

Remark 3. This corollary is motivated by Lemma 5 in [2] which is a crucial step in the proof that if Γ is finite and $q > 1$, then the building corresponding to the Tits system (B,N,W,R) does not have proper quotients satisfying the conditions in [10], Theorem 1. The above corollary shows that the conditions are satisfied for the Chevalley groups of type A_n, D_n or E_m $(6 \leq m \leq 8)$. For another application, see [3].

Remark 4. It is possible to compute the parameters p_{jk}^i for arbitrary k in a similar way. Again one starts by writing $iPkP$ as a disjoint union of sets of the form $iBwP$. Next by induction on $l(w)$ this is rewritten as a disjoint union of cosets uvP, where $u \in U_v^-$ and $v \in L_J$. As an algorithm this works perfectly well, but it is not so easy to give a simple closed expression for p_{jk}^i.

5. Computation in the Weyl group

We shall briefly discuss the way in which several items in the Weyl group have been computed.

(i) The length function l.
The only essential ingredient in our computations is the length function; all other computations could be done by general group theoretic routines. But given the permutation representation of the fundamental reflections on the root system Φ and a product representation $w = s_1 \cdot s_2 \cdots s_m$ (not necessarily minimal), we find $l(w)$ from

$$l(w) = |\{\alpha \epsilon \Phi : \alpha > 0 \text{ and } w\alpha < 0\}|$$

(see e.g. Bourbaki [1] Chap. VI, §1.6 Cor. 2).

(ii) Canonical representatives of the cosets wW_J.
Let Φ be the coroot perpendicular to all fundamental roots except the one corresponding to r. Then Φ has stabilizer W_J in W, and the images of ϕ under W are in 1-1 correspondence with the cosets wW_J.

(iii) Equality in W.
Similarly, let ρ be the sum of all positive roots. Then $w\rho = w'\rho$ iff $w = w'$.

(iv) Double coset representatives.
Given a suitable lexicographic and recursive way of generating the cosets wW_J, the first of these to belong to a certain coset $W_J w W_J$ will have $w \epsilon D_J$. All cosets in the same double coset are found by premultiplying previously found cosets with reflections in J. However, the set D_J of distinguished double coset representatives can be found without listing all single cosets wW_J: given $w \epsilon D_J$, one can determine all elements from $D_J \cap wL$, where $L = L_J \cap W_J r$, by simply sieving all right and left J-reduced words from wL (compare with (i)). In view of the fact that W is generated by $J \cup \{r\}$, iteration of this process will eventually yield all of D_J (one can start with $w = 1$). We have done so for the Weyl groups of type F_4, E_6, E_7, E_8. The cardinalities of D_J, i.e. the permutation ranks, have been given above.

References

[1] N. Bourbaki, *Groupes et Algèbres de Lie,* Chap. 4,5 et 6, Hermann, Paris, 1968.

[2] A.E. Brouwer and A.M. Cohen, "Some remarks on Tits geometries" (with an appendix by J. Tits), *Indagationes Mathematica,* 45 (1983) 392-402.

[3] A.E. Brouwer and A.M. Cohen, "Local recognition of Tits geometries of classical type", preprint.

[4] R.W. Carter, *Simple groups of Lie type*, Wiley, London, 1972.

[5] A.M. Cohen, "Semisimple Lie groups from a geometric viewpoint", in: *The Structure of Real Semisimple Lie Groups* (T.H. Koornwinder, editor) MC Syllabus 49, Math. Centre, Amsterdam, 1982, pp. 41-77.

[6] A.M. Cohen, "Points and lines in metasymplectic spaces", *Annals of Discrete Mathematics* 18 (1983) 193-196.

[7] L. Solomon, "A Mackey formula in the group ring of a Coxeter group", *Journal of Algebra* 41 (1976) 255-268.

[8] J. Tits, "Les "formes réelles" des groupes de type E_6", Séminaire Bourbaki, no. 162, 1958, Paris.

[9] J. Tits, *Buildings of Spherical Type and Finite BN-Pairs*, Lecture Notes in Mathematics 386, Springer, Berlin, 1974.

[10] J. Tits, "A local approach to buildings", in: *The Geometric Vein* (C. Davis et al., editors) Springer, Berlin, 1982, pp. 519-547.

Appendix

$E_{6,1}$
27 cosets
3 double cosets
Sizes:

0: ()
[1] 1
1: (1)
[16] $q + q^2 + q^3 + 2q^4 + 2q^5 + 2q^6 + 2q^7 + 2q^8 + q^9 + q^{10} + q^{11}$
2: (12364321)
[10] $q^8 + q^9 + q^{10} + q^{11} + 2q^{12} + q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 0:
1: [16] $q + q^2 + q^3 + 2q^4 + 2q^5 + 2q^6 + 2q^7 + 2q^8 + q^9 + q^{10} + q^{11}$

Neighbours of a point in 1:
0: [1] 1
1: [10] $-1 + q + q^2 + q^3 + 2q^4 + 2q^5 + 2q^6 + q^7 + q^8$
2: [5] $q^7 + q^8 + q^9 + q^{10} + q^{11}$

Neighbours of a point in 2:

1: [8] $1 + q + q^2 + 2q^3 + q^4 + q^5 + q^6$
2: [8] $-1 - q^3 + q^4 + q^5 + q^6 + 2q^7 + 2q^8 + q^9 + q^{10} + q^{11}$

$E_{6,2}$
216 cosets
10 double cosets
Sizes:
0: ()
[1] 1
1: (2)
[20] $q + 2q^2 + 3q^3 + 4q^4 + 4q^5 + 3q^6 + 2q^7 + q^8$
2: (2312)
[30] $q^4 + 2q^5 + 4q^6 + 5q^7 + 6q^8 + 5q^9 + 4q^{10} + 2q^{11} + q^{12}$
3: (236432)
[10] $q^6 + 2q^7 + 2q^8 + 2q^9 + 2q^{10} + q^{11}$
4: (2364312)
[60] $q^7 + 3q^8 + 6q^9 + 9q^{10} + 11q^{11} + 11q^{12} + 9q^{13} + 6q^{14} + 3q^{15} + q^{16}$
5: (23645342312)
[20] $q^{11} + 2q^{12} + 3q^{13} + 4q^{14} + 4q^{15} + 3q^{16} + 2q^{17} + q^{18}$
6: (23412365432)
[20] $q^{11} + 2q^{12} + 3q^{13} + 4q^{14} + 4q^{15} + 3q^{16} + 2q^{17} + q^{18}$
7: (2341236342312)
[5] $q^{13} + q^{14} + q^{15} + q^{16} + q^{17}$
8: (23412365342312)
[40] $q^{14} + 3q^{15} + 5q^{16} + 7q^{17} + 8q^{18} + 7q^{19} + 5q^{20} + 3q^{21} + q^{22}$
9: (2364534123645342312)
[10] $q^{24} + q^{25}$
Neighbours of a point in 0:
1: [20] $3q^6 + 2q^7 + q^8$
Neighbours of a point in 1:
0: [1] 1
1: [7] $-1 + q + 2q^2 + 2q^3 + 2q^4 + q^5$
2: [6] $q^3 + 2q^4 + 2q^5 + q^6$
3: [3] $q^5 + q^6 + q^7$
4: [3] $q^6 + q^7 + q^8$

Neighbours of a point in 2:

1: [4] $1 + 2q + q^2$
2: [6] $-1 - q + q^2 + 3q^3 + 3q^4 + q^5$
4: [8] $q^4 + 3q^5 + 3q^6 + q^7$
5: [2] $q^7 + q^8$

Neighbours of a point in 3:

1: [6] $1 + q + 2q^2 + q^3 + q^4$
3: [4] $-1 - q^2 + q^3 + q^4 + 2q^5 + q^6 + q^7$
4: [6] $q^2 + q^3 + 2q^4 + q^5 + q^6$
6: [4] $q^5 + q^6 + q^7 + q^8$

Neighbours of a point in 4:

1: [1] 1
2: [4] $q + 2q^2 + q^3$
3: [1] q
4: [7] $-1 - q + 2q^3 + 4q^4 + 3q^5$
5: [2] $q^6 + q^7$
6: [2] $q^5 + q^6$
7: [1] q^6
8: [2] $q^7 + q^8$

Neighbours of a point in 5:

2: [3] $1 + q + q^2$
4: [6] $q^2 + 2q^3 + 2q^4 + q^5$
5: [4] $-1 + q^3 + q^4 + q^5 + q^6 + q^7$
8: [6] $q^4 + 2q^5 + 2q^6 + q^7$
9: [1] q^8

Neighbours of a point in 6:

3: [2] $1 + q$
4: [6] $q + 2q^2 + 2q^3 + q^4$
6: [6] $-1 - q + q^3 + 3q^4 + 3q^5 + q^6$
8: [6] $q^5 + 2q^6 + 2q^7 + q^8$

Neighbours of a point in 7:

4: [12] $1 + 2q + 3q^2 + 3q^3 + 2q^4 + q^5$
7: [0] $-1 - q - q^2 + q^4 + q^5 + q^6$
8: [8] $q^4 + 2q^5 + 2q^6 + 2q^7 + q^8$

Neighbours of a point in 8:

4: [3] $1 + q + q^2$
5: [3] $q + q^2 + q^3$
6: [3] $q^2 + q^3 + q^4$
7: [1] q^3
8: [7] $-1 - q - q^2 + 3q^4 + 4q^5 + 2q^6 + q^7$
9: [3] $q^6 + q^7 + q^8$

Neighbours of a point in 9:

5: [2] $1 + q$
8: [12] $q + 3q^2 + 4q^3 + 3q^4 + q^5$
9: [6] $-1 - q - q^2 - q^3 + q^4 + 3q^5 + 3q^6 + 2q^7 + q^8$

$E_{6,6}$

72 cosets
5 double cosets
Sizes:

0: ()
[1] 1

1: (6)
[20] $q + q^2 + 2q^3 + 3q^4 + 3q^5 + 3q^6 + 3q^7 + 2q^8 + q^9 + q^{10}$

2: (634236)
[30] $q^6 + 2q^7 + 3q^8 + 4q^9 + 5q^{10} + 5q^{11} + 4q^{12} + 3q^{13} + 2q^{14} + q^{15}$

3: (63452341236)
[20] $q^{11} + q^{12} + 2q^{13} + 3q^{14} + 3q^{15} + 3q^{16} + 3q^{17} + 2q^{18} + q^{19} + q^{20}$

4: (6345234123634523412З6)
[1] q^{21}

Neighbours of a point in 0:

1: [20] $q + q^2 + 2q^3 + 3q^4 + 3q^5 + 3q^6 + 3q^7 + 2q^8 + q^9 + q^{10}$

Neighbours of a point in 1:

0: [1] 1
1: [9] $-1 + q + q^2 + 2q^3 + 3q^4 + 2q^5 + q^6$
2: [9] $q^5 + 2q^6 + 3q^7 + 2q^8 + q^9$
3: [1] q^{10}

Neighbours of a point in 2:

1: [6] $1 + q + 2q^2 + q^3 + q^4$
2: [8] $-1 - q^2 + q^3 + 2q^4 + 3q^5 + 2q^6 + 2q^7$

3: [6] $q^6 + q^7 + 2q^8 + q^9 + q^{10}$

Neighbours of a point in 3:

1: [1] 1

2: [9] $q + 2q^2 + 3q^3 + 2q^4 + q^5$

3: [9] $-1 - q^2 - q^3 + q^4 + 2q^5 + 3q^6 + 3q^7 + 2q^8 + q^9$

4: [1] q^{10}

Neighbours of a point in 4:

3: [20] $1 + q + 2q^2 + 3q^3 + 3q^4 + 3q^5 + 3q^6 + 2q^7 + q^8 + q^9$

4: [0] $-1 - q^2 - q^3 + q^7 + q^8 + q^{10}$

$E_{7,1}$

56 cosets
4 double cosets
Sizes:

0: ()

[1] 1

1: (1)

[27] $q + q^2 + q^3 + q^4 + 2q^5 + 2q^6 + 2q^7 + 2q^8 + 3q^9 + 2q^{10} + 2q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16} + q^{17}$

2: (1234754321)

[27] $q^{10} + q^{11} + q^{12} + q^{13} + 2q^{14} + 2q^{15} + 2q^{16} + 2q^{17} + 3q^{18} + 2q^{19} + 2q^{20} + 2q^{21} + 2q^{22} + q^{23} + q^{24} + q^{25} + q^{26}$

3: (123475645347234512347654321)

[1] q^{27}

Neighbours of a point in 0:

1: [27] $q + q^2 + q^3 + q^4 + 2q^5 + 2q^6 + 2q^7 + 2q^8 + 3q^9 + 2q^{10} + 2q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16} + q^{17}$

Neighbours of a point in 1:

0: [1] 1

1: [16] $-1 + q + q^2 + q^3 + q^4 + 2q^5 + 2q^6 + 2q^7 + 2q^8 + 2q^9 + q^{10} + q^{11} + q^{12}$

2: [10] $q^9 + q^{10} + q^{11} + q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16} + q^{17}$

Neighbours of a point in 2:

1: [10] $1 + q + q^2 + q^3 + 2q^4 + q^5 + q^6 + q^7 + q^8$

2: [16] $-1 - q^4 + q^5 + q^6 + q^7 + q^8 + 3q^9 + 2q^{10} + 2q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16}$

3: [1] q^{17}

Neighbours of a point in 3:

2: [27] $1 + q + q^2 + q^3 + 2q^4 + 2q^5 + 2q^6 + 2q^7 + 3q^8 + 2q^9 + 2q^{10} + 2q^{11} + 2q^{12} + q^{13} + q^{14} + q^{15} + q^{16}$

3: [0] $-1 - q^4 - q^8 + q^9 + q^{13} + q^{17}$

$E_{7,6}$

126 cosets

5 double cosets

Sizes:

0: ()

[1] 1

1: (6)

[32] $q + q^2 + q^3 + 2q^4 + 2q^5 + 3q^6 + 3q^7 + 3q^8 + 3q^9 + 3q^{10} + 3q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16}$

2: (65473456)

[60] $q^8 + q^9 + 2q^{10} + 2q^{11} + 4q^{12} + 4q^{13} + 5q^{14} + 5q^{15} + 6q^{16} + 6q^{17} + 5q^{18} + 5q^{19} + 4q^{20} + 4q^{21} + 2q^{22} + 2q^{23} + q^{24} + q^{25}$

3: (65473452347123456)

[32] $q^{17} + q^{18} + q^{19} + 2q^{20} + 2q^{21} + 3q^{22} + 3q^{23} + 3q^{24} + 3q^{25} + 3q^{26} + 3q^{27} + 2q^{28} + 2q^{29} + q^{30} + q^{31} + q^{32}$

4: (654734562345123474563452347123456)

[1] q^{33}

Neighbours of a point in 0:

1: [32] $q + q^2 + q^3 + 2q^4 + 2q^5 + 3q^6 + 3q^7 + 3q^8 + 3q^9 + 3q^{10} + 3q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 1:

0: [1] 1

1: [15] $-1 + q + q^2 + q^3 + 2q^4 + 2q^5 + 3q^6 + 2q^7 + 2q^8 + q^9 + q^{10}$

2: [15] $q^7 + q^8 + 2q^9 + 2q^{10} + 3q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15}$

3: [1] q^{16}

Neighbours of a point in 2:

1: [8] $1 + q + q^2 + 2q^3 + q^4 + q^5 + q^6$

2: [16] $-1 - q^3 + q^4 + q^5 + 2q^6 + 3q^7 + 3q^8 + 3q^9 + 2q^{10} + 2q^{11} + q^{12}$

3: [8] $q^{10} + q^{11} + q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 3:

1: [1] 1

2: [15] $q + q^2 + 2q^3 + 2q^4 + 3q^5 + 2q^6 + 2q^7 + q^8 + q^9$

3: [15] $-1 - q^3 - q^5 + q^6 + q^7 + 2q^8 + 2q^9 + 3q^{10} + 3q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15}$

4: [1] q^{16}

Neighbours of a point in 4:

3: [32] $1 + q + q^2 + 2q^3 + 2q^4 + 3q^5 + 3q^6 + 3q^7 + 3q^8 + 3q^9 + 3q^{10} + 2q^{11} + 2q^{12} + q^{13} + q^{14} + q^{15}$

4: [0] $-1 - q^3 - q^5 + q^{11} + q^{13} + q^{16}$

$E_{7,7}$

576 cosets
10 double cosets
Sizes:

0: ()
[1] 1

1: (7)
[35] $q + q^2 + 2q^3 + 3q^4 + 4q^5 + 4q^6 + 5q^7 + 4q^8 + 4q^9 + 3q^{10} + 2q^{11} + q^{12} + q^{13}$

2: (745347)
[105] $q^6 + 2q^7 + 4q^8 + 6q^9 + 9q^{10} + 11q^{11} + 13q^{12} + 13q^{13} + 13q^{14} + 11q^{15} + 9q^{16} + 6q^{17} + 4q^{18} + 2q^{19} + q^{20}$

3: (74563452347)
[140] $q^{11} + 2q^{12} + 4q^{13} + 7q^{14} + 10q^{15} + 13q^{16} + 16q^{17} + 17q^{18} + 17q^{19} + 16q^{20} + 13q^{21} + 10q^{22} + 7q^{23} + 4q^{24} + 2q^{25} + q^{26}$

4: (745347234512347)
[7] $q^{15} + q^{16} + q^{17} + q^{18} + q^{19} + q^{20} + q^{21}$

5: (7453476234512347)
[140] $q^{16} + 2q^{17} + 4q^{18} + 7q^{19} + 10q^{20} + 13q^{21} + 16q^{22} + 17q^{23} + 17q^{24} + 16q^{25} + 13q^{26} + 10q^{27} + 7q^{28} + 4q^{29} + 2q^{30} + q^{31}$

6: (74563452347456345234 7)
[7] $q^{21} + q^{22} + q^{23} + q^{24} + q^{25} + q^{26} + q^{27}$

7: (745634523474563451234 7)
[105] $q^{22} + 2q^{23} + 4q^{24} + 6q^{25} + 9q^{26} + 11q^{27} + 13q^{28} + 13q^{29} + 13q^{30} + 11q^{31} + 9q^{32} + 6q^{33} + 4q^{34} + 2q^{35} + q^{36}$

8: (745347623451234734562345123 47)

[35] $q^{29} + q^{30} + 2q^{31} + 3q^{32} + 4q^{33} + 4q^{34} + 5q^{35} + 4q^{36} + 4q^{37} + 3q^{38} + 2q^{39} + q^{40} + q^{41}$

9: (745347623451234734562345123473456234512347)

[1] q^{42}

Neighbours of a point in 0:

1: [35] $q + q^2 + 2q^3 + 3q^4 + 4q^5 + 4q^6 + 5q^7 + 4q^8 + 4q^9 + 3q^{10} + 2q^{11} + q^{12} + q^{13}$

Neighbours of a point in 1:

0: [1] 1

1: [12] $-1 + q + q^2 + 2q^3 + 3q^4 + 3q^5 + 2q^6 + q^7$

2: [18] $q^5 + 2q^6 + 4q^7 + 4q^8 + 4q^9 + 2q^{10} + q^{11}$

3: [4] $q^{10} + q^{11} + q^{12} + q^{13}$

Neighbours of a point in 2:

1: [6] $1 + q + 2q^2 + q^3 + q^4$

2: [12] $-1 - q^2 + q^3 + 2q^4 + 4q^5 + 3q^6 + 3q^7 + q^8$

3: [12] $q^6 + 2q^7 + 3q^8 + 3q^9 + 2q^{10} + q^{11}$

4: [1] q^9

5: [4] $q^{10} + q^{11} + q^{12} + q^{13}$

Neighbours of a point in 3:

1: [1] 1

2: [9] $q + 2q^2 + 3q^3 + 2q^4 + q^5$

3: [12] $-1 - q^2 - q^3 + q^4 + 3q^5 + 4q^6 + 4q^7 + 2q^8 + q^9$

5: [9] $q^7 + 2q^8 + 3q^9 + 2q^{10} + q^{11}$

6: [1] q^{10}

7: [3] $q^{11} + q^{12} + q^{13}$

Neighbours of a point in 4:

2: [15] $1 + q + 2q^2 + 2q^3 + 3q^4 + 2q^5 + 2q^6 + q^7 + q^8$

4: [0] $-1 - q^2 - q^4 + q^5 + q^7 + q^9$

5: [20] $q^4 + q^5 + 2q^6 + 3q^7 + 3q^8 + 3q^9 + 3q^{10} + 2q^{11} + q^{12} + q^{13}$

Neighbours of a point in 5:

2: [3] $1 + q + q^2$

3: [9] $q^2 + 2q^3 + 3q^4 + 2q^5 + q^6$

4: [1] q^3

5: [12] $-1 - q^2 - q^3 + 2q^5 + 3q^6 + 5q^7 + 3q^8 + 2q^9$

7: [9] $q^8 + 2q^9 + 3q^{10} + 2q^{11} + q^{12}$

8: [1] q^{13}

Neighbours of a point in 6:

3: [20] $1 + q + 2q^2 + 3q^3 + 3q^4 + 3q^5 + 3q^6 + 2q^7 + q^8 + q^9$

6: [0] $-1 - q^2 - q^3 + q^7 + q^8 + q^{10}$

7: [15] $q^5 + q^6 + 2q^7 + 2q^8 + 3q^9 + 2q^{10} + 2q^{11} + q^{12} + q^{13}$

Neighbours of a point in 7:

3: [4] $1 + q + q^2 + q^3$

5: [12] $q^2 + 2q^3 + 3q^4 + 3q^5 + 2q^6 + q^7$

6: [1] q^4

7: [12] $-1 - q^2 - q^3 - q^4 + q^5 + 2q^6 + 4q^7 + 4q^8 + 3q^9 + 2q^{10}$

8: [6] $q^9 + q^{10} + 2q^{11} + q^{12} + q^{13}$

Neighbours of a point in 8:

5: [4] $1 + q + q^2 + q^3$

7: [18] $q^2 + 2q^3 + 4q^4 + 4q^5 + 4q^6 + 2q^7 + q^8$

8: [12] $-1 - q^2 - q^3 - q^4 + 3q^7 + 3q^8 + 4q^9 + 3q^{10} + 2q^{11} + q^{12}$

9: [1] q^{13}

Neighbours of a point in 9:

8: [35] $1 + q + 2q^2 + 3q^3 + 4q^4 + 4q^5 + 5q^6 + 4q^7 + 4q^8 + 3q^9 + 2q^{10} + q^{11} + q^{12}$

9: [0] $-1 - q^2 - q^3 - q^4 - q^6 + q^7 + q^9 + q^{10} + q^{11} + q^{13}$

$E_{8,1}$

240 cosets
5 double cosets
Sizes:

0: ()

[1] 1

1: (1)

[56] $q + q^2 + q^3 + q^4 + q^5 + 2q^6 + 2q^7 + 2q^8 + 2q^9 + 3q^{10} + 3q^{11} + 3q^{12} + 3q^{13} + 3q^{14} + 3q^{15} + 3q^{16} + 3q^{17} + 3q^{18} + 3q^{19} + 2q^{20} + 2q^{21} + 2q^{22} + 2q^{23} + q^{24} + q^{25} + q^{26} + q^{27} + q^{28}$

2: (123458654321)

[126] $q^{12} + q^{13} + q^{14} + q^{15} + 2q^{16} + 2q^{17} + 3q^{18} + 3q^{19} + 4q^{20} + 4q^{21} + 5q^{22} + 5q^{23} + 6q^{24} + 6q^{25} + 6q^{26} + 6q^{27} + 7q^{28} + 7q^{29} + 6q^{30} + 6q^{31} + 6q^{32} + 6q^{33} + 5q^{34} + 5q^{35} + 4q^{36} + 4q^{37} + 3q^{38} + 3q^{39} + 2q^{40} + 2q^{41} + q^{42} + q^{43} + q^{44} + q^{45}$

3: (12345867564583456234587654321)

[56] $q^{29} + q^{30} + q^{31} + q^{32} + q^{33} + 2q^{34} + 2q^{35} + 2q^{36} + 2q^{37} + 3q^{38} + 3q^{39} + 3q^{40} + 3q^{41} + 3q^{42} + 3q^{43} + 3q^{44} + 3q^{45} + 3q^{46} + 3q^{47} + 2q^{48} + 2q^{49} + 2q^{50} + 2q^{51} + q^{52} + q^{53} + q^{54} + q^{55} + q^{56}$

4: (123458675645834567234561234585674563458234561234587654321)

[1] q^{57}

Neighbours of a point in 0:

1: [56] $q + q^2 + q^3 + q^4 + q^5 + 2q^6 + 2q^7 + 2q^8 + 2q^9 + 3q^{10} + 3q^{11} + 3q^{12} + 3q^{13} + 3q^{14} + 3q^{15} + 3q^{16} + 3q^{17} + 3q^{18} + 3q^{19} + 2q^{20} + 2q^{21} + 2q^{22} + 2q^{23} + q^{24} + q^{25} + q^{26} + q^{27} + q^{28}$

Neighbours of a point in 1:

0: [1] 1

1: [27] $-1 + q + q^2 + q^3 + q^4 + q^5 + 2q^6 + 2q^7 + 2q^8 + 2q^9 + 3q^{10} + 2q^{11} + 2q^{12} + 2q^{13} + 2q^{14} + q^{15} + q^{16} + q^{17} + q^{18}$

2: [27] $q^{11} + q^{12} + q^{13} + q^{14} + 2q^{15} + 2q^{16} + 2q^{17} + 2q^{18} + 3q^{19} + 2q^{20} + 2q^{21} + 2q^{22} + 2q^{23} + q^{24} + q^{25} + q^{26} + q^{27}$

3: [1] q^{28}

Neighbours of a point in 2:

1: [12] $1 + q + q^2 + q^3 + q^4 + 2q^5 + q^6 + q^7 + q^8 + q^9 + q^{10}$

2: [32] $-1 - q^5 + q^6 + q^7 + q^8 + q^9 + 2q^{10} + 3q^{11} + 3q^{12} + 3q^{13} + 3q^{14} + 3q^{15} + 3q^{16} + 3q^{17} + 2q^{18} + 2q^{19} + q^{20} + q^{21} + q^{22}$

3: [12] $q^{18} + q^{19} + q^{20} + q^{21} + q^{22} + 2q^{23} + q^{24} + q^{25} + q^{26} + q^{27} + q^{28}$

Neighbours of a point in 3:

1: [1] 1

2: [27] $q + q^2 + q^3 + q^4 + 2q^5 + 2q^6 + 2q^7 + 2q^8 + 3q^9 + 2q^{10} + 2q^{11} + 2q^{12} + 2q^{13} + q^{14} + q^{15} + q^{16} + q^{17}$

3: [27] $-1 - q^5 - q^9 + q^{10} + q^{11} + q^{12} + q^{13} + 2q^{14} + 2q^{15} + 2q^{16} + 2q^{17} + 3q^{18} + 3q^{19} + 2q^{20} + 2q^{21} + 2q^{22} + 2q^{23} + q^{24} + q^{25} + q^{26} + q^{27}$

4: [1] q^{28}

Neighbours of a point in 4:

3: [56] $1 + q + q^2 + q^3 + q^4 + 2q^5 + 2q^6 + 2q^7 + 2q^8 + 3q^9 + 3q^{10} + 3q^{11} + 3q^{12} + 3q^{13} + 3q^{14} + 3q^{15} + 3q^{16} + 3q^{17} + 3q^{18} + 2q^{19} + 2q^{20} + 2q^{21} + 2q^{22} + q^{23} + q^{24} + q^{25} + q^{26} + q^{27}$

4: [0] $-1 - q^5 - q^9 + q^{19} + q^{23} + q^{28}$

$E_{8,7}$

2160 cosets
10 double cosets
Sizes:

0: ()

[1] 1

1: (7)

[64] $q + q^{2} + q^{3} + 2q^{4} + 2q^{5} + 3q^{6} + 4q^{7} + 4q^{8} + 4q^{9} + 5q^{10} + 5q^{11} + 5q^{12} + 5q^{13} + 4q^{14} + 4q^{15} + 4q^{16} + 3q^{17} + 2q^{18} + 2q^{19} + q^{20} + q^{21} + q^{22}$

2: (76584567)

[280] $q^{8} + q^{9} + 2q^{10} + 3q^{11} + 5q^{12} + 6q^{13} + 9q^{14} + 10q^{15} + 13q^{16} + 15q^{17} + 17q^{18} + 18q^{19} + 20q^{20} + 20q^{21} + 20q^{22} + 20q^{23} + 18q^{24} + 17q^{25} + 15q^{26} + 13q^{27} + 10q^{28} + 9q^{29} + 6q^{30} + 5q^{31} + 3q^{32} + 2q^{33} + q^{34} + q^{35}$

3: (76584563458234567)

[448] $q^{17} + 2q^{18} + 3q^{19} + 5q^{20} + 7q^{21} + 10q^{22} + 14q^{23} + 17q^{24} + 20q^{25} + 24q^{26} + 27q^{27} + 30q^{28} + 32q^{29} + 32q^{30} + 32q^{31} + 32q^{32} + 30q^{33} + 27q^{34} + 24q^{35} + 20q^{36} + 17q^{37} + 14q^{38} + 10q^{39} + 7q^{40} + 5q^{41} + 3q^{42} + 2q^{43} + q^{44}$

4: (765845673456234581234567)

[560] $q^{24} + q^{25} + 2q^{26} + 4q^{27} + 6q^{28} + 8q^{29} + 12q^{30} + 15q^{31} + 19q^{32} + 24q^{33} + 27q^{34} + 31q^{35} + 35q^{36} + 37q^{37} + 38q^{38} + 40q^{39} + 38q^{40} + 37q^{41} + 35q^{42} + 31q^{43} + 27q^{44} + 24q^{45} + 19q^{46} + 15q^{47} + 12q^{48} + 8q^{49} + 6q^{50} + 4q^{51} + 2q^{52} + q^{53} + q^{54}$

5: (765845673456234585674563458234567)

[14] $q^{33} + q^{34} + q^{35} + q^{36} + q^{37} + q^{38} + 2q^{39} + q^{40} + q^{41} + q^{42} + q^{43} + q^{44} + q^{45}$

6: (7658456734562345856745634581234567)

[448] $q^{34} + 2q^{35} + 3q^{36} + 5q^{37} + 7q^{38} + 10q^{39} + 14q^{40} + 17q^{41} + 20q^{42} + 24q^{43} + 27q^{44} + 30q^{45} + 32q^{46} + 32q^{47} + 32q^{48} + 32q^{49} + 30q^{50} + 27q^{51} + 24q^{52} + 20q^{53} + 17q^{54} + 14q^{55} + 10q^{56} + 7q^{57} + 5q^{58} + 3q^{59} + 2q^{60} + q^{61}$

7: (7658456345872345612345845673456234581234567)

[280] $q^{43} + q^{44} + 2q^{45} + 3q^{46} + 5q^{47} + 6q^{48} + 9q^{49} + 10q^{50} + 13q^{51} + 15q^{52} + 17q^{53} + 18q^{54} + 20q^{55} + 20q^{56} + 20q^{57} + 20q^{58} + 18q^{59} + 17q^{60} + 15q^{61} + 13q^{62} + 10q^{63} + 9q^{64} + 6q^{65} + 5q^{66} + 3q^{67} + 2q^{68} + q^{69} + q^{70}$

8: (76584563458723456123458456734562345845673456234581234567)

[64] $q^{56} + q^{57} + q^{58} + 2q^{59} + 2q^{60} + 3q^{61} + 4q^{62} + 4q^{63} + 4q^{64} + 5q^{65} + 5q^{66} + 5q^{67} + 5q^{68} + 4q^{69} + 4q^{70} + 4q^{71} + 3q^{72} + 2q^{73} + 2q^{74} + q^{75} + q^{76} + q^{77}$

9: (765845673456234585674561234586723456123458345672345612345845
673456234581234567)

[1] q^{78}

Neighbours of a point in 0:

1: [64] $q + q^2 + q^3 + 2q^4 + 2q^5 + 3q^6 + 4q^7 + 4q^8 + 4q^9 + 5q^{10} + 5q^{11} + 5q^{12} + 5q^{13} + 4q^{14} + 4q^{15} + 4q^{16} + 3q^{17} + 2q^{18} + 2q^{19} + q^{20} + q^{21} + q^{22}$

Neighbours of a point in 1:

0: [1] 1

1: [21] $-1 + q + q^2 + q^3 + 2q^4 + 2q^5 + 3q^6 + 3q^7 + 3q^8 + 2q^9 + 2q^{10} + q^{11} + q^{12}$

2: [35] $q^7 + q^8 + 2q^9 + 3q^{10} + 4q^{11} + 4q^{12} + 5q^{13} + 4q^{14} + 4q^{15} + 3q^{16} + 2q^{17} + q^{18} + q^{19}$

3: [7] $q^{16} + q^{17} + q^{18} + q^{19} + q^{20} + q^{21} + q^{22}$

Neighbours of a point in 2:

1: [8] $1 + q + q^2 + 2q^3 + q^4 + q^5 + q^6$

2: [24] $-1 - q^3 + q^4 + q^5 + 2q^6 + 4q^7 + 4q^8 + 4q^9 + 4q^{10} + 3q^{11} + 2q^{12} + q^{13}$

3: [24] $q^{10} + 2q^{11} + 3q^{12} + 4q^{13} + 4q^{14} + 4q^{15} + 3q^{16} + 2q^{17} + q^{18}$

4: [8] $q^{16} + q^{17} + q^{18} + 2q^{19} + q^{20} + q^{21} + q^{22}$

Neighbours of a point in 3:

1: [1] 1

2: [15] $q + q^2 + 2q^3 + 2q^4 + 3q^5 + 2q^6 + 2q^7 + q^8 + q^9$

3: [21] $-1 - q^3 - q^5 + q^6 + 2q^7 + 3q^8 + 3q^9 + 4q^{10} + 4q^{11} + 3q^{12} + 2q^{13} + q^{14} + q^{15}$

4: [20] $q^{10} + q^{11} + 2q^{12} + 3q^{13} + 3q^{14} + 3q^{15} + 3q^{16} + 2q^{17} + q^{18} + q^{19}$

5: [1] q^{16}

6: [6] $q^{17} + q^{18} + q^{19} + q^{20} + q^{21} + q^{22}$

Neighbours of a point in 4:

2: [4] $1 + q + q^2 + q^3$

3: [16] $q^3 + 2q^4 + 3q^5 + 4q^6 + 3q^7 + 2q^8 + q^9$

4: [24] $-1 - q^3 - q^5 - q^6 + q^7 + 2q^8 + 3q^9 + 5q^{10} + 5q^{11} + 5q^{12} + 4q^{13} + 2q^{14} + q^{15}$

6: [16] $q^{13} + 2q^{14} + 3q^{15} + 4q^{16} + 3q^{17} + 2q^{18} + q^{19}$

7: [4] $q^{19} + q^{20} + q^{21} + q^{22}$

Neighbours of a point in 5:

3: [32] $1 + q + q^2 + 2q^3 + 2q^4 + 3q^5 + 3q^6 + 3q^7 + 3q^8 + 3q^9 + 3q^{10} + 2q^{11} + 2q^{12} + q^{13} + q^{14} + q^{15}$

5: [0] $-1 - q^3 - q^5 + q^{11} + q^{13} + q^{16}$

6: [32] $q^7 + q^8 + q^9 + 2q^{10} + 2q^{11} + 3q^{12} + 3q^{13} + 3q^{14} + 3q^{15} + 3q^{16} + 3q^{17} + 2q^{18} + 2q^{19} + q^{20} + q^{21} + q^{22}$

Neighbours of a point in 6:

3: [6] $1 + q + q^2 + q^3 + q^4 + q^5$

4: [20] $q^3 + q^4 + 2q^5 + 3q^6 + 3q^7 + 3q^8 + 3q^9 + 2q^{10} + q^{11} + q^{12}$

5: [1] q^6

6: [21] $-1 - q^3 - q^5 - q^6 + q^7 + q^8 + q^9 + 3q^{10} + 4q^{11} + 4q^{12} + 4q^{13} + 3q^{14} + 2q^{15} + 2q^{16}$

7: [15] $q^{13} + q^{14} + 2q^{15} + 2q^{16} + 3q^{17} + 2q^{18} + 2q^{19} + q^{20} + q^{21}$

8: [1] q^{22}

Neighbours of a point in 7:

4: [8] $1 + q + q^2 + 2q^3 + q^4 + q^5 + q^6$

6: [24] $q^4 + 2q^5 + 3q^6 + 4q^7 + 4q^8 + 4q^9 + 3q^{10} + 2q^{11} + q^{12}$

7: [24] $-1 - q^3 - q^5 - q^6 + 2q^{10} + 3q^{11} + 4q^{12} + 5q^{13} + 4q^{14} + 4q^{15} + 3q^{16} + 2q^{17} + q^{18}$

8: [8] $q^{16} + q^{17} + q^{18} + 2q^{19} + q^{20} + q^{21} + q^{22}$

Neighbours of a point in 8:

6: [7] $1 + q + q^2 + q^3 + q^4 + q^5 + q^6$

7: [35] $q^3 + q^4 + 2q^5 + 3q^6 + 4q^7 + 4q^8 + 5q^9 + 4q^{10} + 4q^{11} + 3q^{12} + 2q^{13} + q^{14} + q^{15}$

8: [21] $-1 - q^3 - q^5 - q^6 - q^9 + q^{10} + q^{11} + 2q^{12} + 3q^{13} + 3q^{14} + 3q^{15} + 4q^{16} + 3q^{17} + 2q^{18} + 2q^{19} + q^{20} + q^{21}$

9: [1] q^{22}

Neighbours of a point in 9:

8: [64] $1 + q + q^2 + 2q^3 + 2q^4 + 3q^5 + 4q^6 + 4q^7 + 4q^8 + 5q^9 + 5q^{10} + 5q^{11} + 5q^{12} + 4q^{13} + 4q^{14} + 4q^{15} + 3q^{16} + 2q^{17} + 2q^{18} + q^{19} + q^{20} + q^{21}$

9: [0] $-1 - q^3 - q^5 - q^6 - q^9 + q^{13} + q^{16} + q^{17} + q^{19} + q^{22}$

$E_{8,8}$

17280 cosets

35 double cosets

Sizes:

0: ()

[1] 1

1: (8)

[56] $q + q^2 + 2q^3 + 3q^4 + 4q^5 + 5q^6 + 6q^7 + 6q^8 + 6q^9 + 6q^{10} + 5q^{11} + 4q^{12} + 3q^{13} + 2q^{14} + q^{15} + q^{16}$

2: (856458)

[280] $q^6 + 2q^7 + 4q^8 + 7q^9 + 11q^{10} + 15q^{11} + 20q^{12} + 24q^{13} + 27q^{14} + 29q^{15} + 29q^{16} + 27q^{17} + 24q^{18} + 20q^{19} + 15q^{20} + 11q^{21} + 7q^{22} + 4q^{23} + 2q^{24} + q^{25}$

3: (85674563458)

[560] $q^{11} + 2q^{12} + 5q^{13} + 9q^{14} + 15q^{15} + 22q^{16} + 31q^{17} + 39q^{18} + 47q^{19} + 53q^{20} + 56q^{21} + 56q^{22} + 53q^{23} + 47q^{24} + 39q^{25} + 31q^{26} + 22q^{27} + 15q^{28} + 9q^{29} + 5q^{30} + 2q^{31} + q^{32}$

4: (856458345623458)

[56] $q^{15} + 2q^{16} + 3q^{17} + 4q^{18} + 5q^{19} + 6q^{20} + 7q^{21} + 7q^{22} + 6q^{23} + 5q^{24} + 4q^{25} + 3q^{26} + 2q^{27} + q^{28}$

5: (8564587345623458)

[1120] $q^{16} + 3q^{17} + 7q^{18} + 14q^{19} + 24q^{20} + 37q^{21} + 53q^{22} + 70q^{23} + 86q^{24} + 100q^{25} + 109q^{26} + 112q^{27} + 109q^{28} + 100q^{29} + 86q^{30} + 70q^{31} + 53q^{32} + 37q^{33} + 24q^{34} + 14q^{35} + 7q^{36} + 3q^{37} + q^{38}$

6: (8567456345856745663458)

[28] $q^{21} + q^{22} + 2q^{23} + 2q^{24} + 3q^{25} + 3q^{26} + 4q^{27} + 3q^{28} + 3q^{29} + 2q^{30} + 2q^{31} + q^{32} + q^{33}$

7: (85645834567234561234 58)

[280] $q^{22} + 2q^{23} + 4q^{24} + 7q^{25} + 11q^{26} + 15q^{27} + 20q^{28} + 24q^{29} + 27q^{30} + 29q^{31} + 29q^{32} + 27q^{33} + 24q^{34} + 20q^{35} + 15q^{36} + 11q^{37} + 7q^{38} + 4q^{39} + 2q^{40} + q^{41}$

8: (8567456345823456123458)

[280] $q^{22} + 2q^{23} + 4q^{24} + 7q^{25} + 11q^{26} + 15q^{27} + 20q^{28} + 24q^{29} + 27q^{30} + 29q^{31} + 29q^{32} + 27q^{33} + 24q^{34} + 20q^{35} + 15q^{36} + 11q^{37} + 7q^{38} + 4q^{39} + 2q^{40} + q^{41}$

9: (856745634585674562 3458)

[840] $q^{22} + 3q^{23} + 7q^{24} + 13q^{25} + 22q^{26} + 33q^{27} + 46q^{28} + 59q^{29} + 71q^{30} + 80q^{31} + 85q^{32} + 85q^{33} + 80q^{34} + 71q^{35} + 59q^{36} + 46q^{37} + 33q^{38} + 22q^{39} + 13q^{40} + 7q^{41} + 3q^{42} + q^{43}$

10: (8567456345856723456123458)

[1680] $q^{25} + 3q^{26} + 8q^{27} + 16q^{28} + 29q^{29} + 46q^{30} + 68q^{31} + 92q^{32} + 117q^{33} + 139q^{34} + 156q^{35} + 165q^{36} + 165q^{37} + 156q^{38} + 139q^{39} + 117q^{40} + 92q^{41} + 68q^{42} + 46q^{43} + 29q^{44} + 16q^{45} + 8q^{46} + 3q^{47} + q^{48}$

11: (85645873456234584567345623458)

[280] $q^{29} + 2q^{30} + 4q^{31} + 7q^{32} + 11q^{33} + 15q^{34} + 20q^{35} + 24q^{36} + 27q^{37} + 29q^{38} + 29q^{39} + 27q^{40} + 24q^{41} + 20q^{42} + 15q^{43} + 11q^{44} + 7q^{45} + 4q^{46} + 2q^{47} + q^{48}$

12: (856458734562345845673456123458)

[1680] $q^{30} + 3q^{31} + 8q^{32} + 16q^{33} + 29q^{34} + 46q^{35} + 68q^{36} + 92q^{37} + 117q^{38} + 139q^{39} + 156q^{40} + 165q^{41} + 165q^{42} + 156q^{43} + 139q^{44} + 117q^{45} + 92q^{46} + 68q^{47} + 46q^{48} + 29q^{49} + 16q^{50} + 8q^{51} + 3q^{52} + q^{53}$

13: (856745634585674563458234561234 58)

[168] $q^{32} + 2q^{33} + 4q^{34} + 6q^{35} + 9q^{36} + 12q^{37} + 15q^{38} + 17q^{39} + 18q^{40} + 18q^{41} + 17q^{42} + 15q^{43} + 12q^{44} + 9q^{45} + 6q^{46} + 4q^{47} + 2q^{48} + q^{49}$

14: (856458345672345612345856745634 58)

[168] $q^{32} + 2q^{33} + 4q^{34} + 6q^{35} + 9q^{36} + 12q^{37} + 15q^{38} + 17q^{39} + 18q^{40} + 18q^{41} + 17q^{42} + 15q^{43} + 12q^{44} + 9q^{45} + 6q^{46} + 4q^{47} + 2q^{48} + q^{49}$

15: (856745634585674562345867234561234 58)

[1120] $q^{35} + 3q^{36} + 7q^{37} + 14q^{38} + 24q^{39} + 37q^{40} + 53q^{41} + 70q^{42} + 86q^{43} + 100q^{44} + 109q^{45} + 112q^{46} + 109q^{47} + 100q^{48} + 86q^{49} + 70q^{50} + 53q^{51} + 37q^{52} + 24q^{53} + 14q^{54} + 7q^{55} + 3q^{56} + q^{57}$

16: (856458345672345612345845673456234 58)

[1120] $q^{35} + 3q^{36} + 7q^{37} + 14q^{38} + 24q^{39} + 37q^{40} + 53q^{41} + 70q^{42} + 86q^{43} + 100q^{44} + 109q^{45} + 112q^{46} + 109q^{47} + 100q^{48} + 86q^{49} + 70q^{50} + 53q^{51} + 37q^{52} + 24q^{53} + 14q^{54} + 7q^{55} + 3q^{56} + q^{57}$

17: (85674563458234561234583456723456123458)

[70] $q^{38} + q^{39} + 2q^{40} + 3q^{41} + 5q^{42} + 5q^{43} + 7q^{44} + 7q^{45} + 8q^{46} + 7q^{47} + 7q^{48} + 5q^{49} + 5q^{50} + 3q^{51} + 2q^{52} + q^{53} + q^{54}$

18: (85674563458567234561234585672345612345 8)

[1680] $q^{39} + 3q^{40} + 8q^{41} + 16q^{42} + 29q^{43} + 46q^{44} + 68q^{45} + 92q^{46} + 117q^{47} + 139q^{48} + 156q^{49} + 165q^{50} + 165q^{51} + 156q^{52} + 139q^{53} + 117q^{54} + 92q^{55} + 68q^{56} + 46q^{57} + 29q^{58} + 16q^{59} + 8q^{60} + 3q^{61} + q^{62}$

19: (8564587345623458456734562345845673456234 58)

[8] $q^{42} + q^{43} + q^{44} + q^{45} + q^{46} + q^{47} + q^{48} + q^{49}$

20: (8564583456723456123458567456345823456123458)

[8] $q^{43} + q^{44} + q^{45} + q^{46} + q^{47} + q^{48} + q^{49} + q^{50}$

21: (8564587345623458456734562345845673456123458)

[168] $q^{43} + 2q^{44} + 4q^{45} + 6q^{46} + 9q^{47} + 12q^{48} + 15q^{49} + 17q^{50} + 18q^{51} + 18q^{52} + 17q^{53} + 15q^{54} + 12q^{55} + 9q^{56} + 6q^{57} + 4q^{58} + 2q^{59} + q^{60}$

22: (8564587345623458456734561234584567345623458)

[168] $q^{43} + 2q^{44} + 4q^{45} + 6q^{46} + 9q^{47} + 12q^{48} + 15q^{49} + 17q^{50} + 18q^{51} + 18q^{52} + 17q^{53} + 15q^{54} + 12q^{55} + 9q^{56} + 6q^{57} + 4q^{58} + 2q^{59} + q^{60}$

23: (856458734562345845673456123458456734561234 58)

[1680] $q^{44} + 3q^{45} + 8q^{46} + 16q^{47} + 29q^{48} + 46q^{49} + 68q^{50} + 92q^{51} + 117q^{52} + 139q^{53} + 156q^{54} + 165q^{55} + 165q^{56} + 156q^{57} + 139q^{58} + 117q^{59} + 92q^{60} + 68q^{61} + 46q^{62} + 29q^{63} + 16q^{64} + 8q^{65} + 3q^{66} + q^{67}$

24: (85645834567234561234585674563458723456123458)

[280] $q^{44} + 2q^{45} + 4q^{46} + 7q^{47} + 11q^{48} + 15q^{49} + 20q^{50} + 24q^{51} + 27q^{52} + 29q^{53} + 29q^{54} + 27q^{55} + 24q^{56} + 20q^{57} + 15q^{58} + 11q^{59} + 7q^{60} + 4q^{61} + 2q^{62} + q^{63}$

25: (856458345672345612345845673456234585672345612 3458)

[840] $q^{49} + 3q^{50} + 7q^{51} + 13q^{52} + 22q^{53} + 33q^{54} + 46q^{55} + 59q^{56} + 71q^{57} + 80q^{58} + 85q^{59} + 85q^{60} + 80q^{61} + 71q^{62} + 59q^{63} + 46q^{64} + 33q^{65} + 22q^{66} + 13q^{67} + 7q^{68} + 3q^{69} + q^{70}$

26: (856745634585674562345867234561234583456723456123458)

[280] $q^{51} + 2q^{52} + 4q^{53} + 7q^{54} + 11q^{55} + 15q^{56} + 20q^{57} + 24q^{58} + 27q^{59} + 29q^{60} + 29q^{61} + 27q^{62} + 24q^{63} + 20q^{64} + 15q^{65} + 11q^{66} + 7q^{67} + 4q^{68} + 2q^{69} + q^{70}$

27: (856745634582345612345834567234561234584567345623458)

[280] $q^{51} + 2q^{52} + 4q^{53} + 7q^{54} + 11q^{55} + 15q^{56} + 20q^{57} + 24q^{58} + 27q^{59} + 29q^{60} + 29q^{61} + 27q^{62} + 24q^{63} + 20q^{64} + 15q^{65} + 11q^{66} + 7q^{67} + 4q^{68} + 2q^{69} + q^{70}$

28: (85674563458567234561234585672345612345845672345612 3458)

[1120] $q^{54} + 3q^{55} + 7q^{56} + 14q^{57} + 24q^{58} + 37q^{59} + 53q^{60} + 70q^{61} + 86q^{62} + 100q^{63} + 109q^{64} + 112q^{65} + 109q^{66} + 100q^{67} + 86q^{68} + 70q^{69} + 53q^{70} + 37q^{71} + 24q^{72} + 14q^{73} + 7q^{74} + 3q^{75} + q^{76}$

29: (8564587345623458456734561234584567345623458345672345612 3458)

[28] $q^{59} + q^{60} + 2q^{61} + 2q^{62} + 3q^{63} + 3q^{64} + 4q^{65} + 3q^{66} + 3q^{67} + 2q^{68} + 2q^{69} + q^{70} + q^{71}$

30: (856458734562345845673456123458456734561234583456723456123458)

[560] $q^{60} + 2q^{61} + 5q^{62} + 9q^{63} + 15q^{64} + 22q^{65} + 31q^{66} + 39q^{67} + 47q^{68} + 53q^{69} + 56q^{70} + 56q^{71} + 53q^{72} + 47q^{73} + 39q^{74} + 31q^{75} + 22q^{76} + 15q^{77} + 9q^{78} + 5q^{79} + 2q^{80} + q^{81}$

31: (85674563458567456234586723456123458345672345612345845673456 23458)

[56] $q^{64} + 2q^{65} + 3q^{66} + 4q^{67} + 5q^{68} + 6q^{69} + 7q^{70} + 7q^{71} + 6q^{72} + 5q^{73} + 4q^{74} + 3q^{75} + 2q^{76} + q^{77}$

32:
(8567456345856745623458672345612345834567234561234583456723456123458)

[280] $q^{67} + 2q^{68} + 4q^{69} + 7q^{70} + 11q^{71} + 15q^{72} + 20q^{73} + 24q^{74} + 27q^{75} + 29q^{76} + 29q^{77} + 27q^{78} + 24q^{79} + 20q^{80} + 15q^{81} + 11q^{82} + 7q^{83} + 4q^{84} + 2q^{85} + q^{86}$

33:
(85645873456234584567345612345845673456123458345672345612345834567
23456123458)

[56] $q^{76} + q^{77} + 2q^{78} + 3q^{79} + 4q^{80} + 5q^{81} + 6q^{82} + 6q^{83} + 6q^{84} + 6q^{85} + 5q^{86} + 4q^{87} + 3q^{88} + 2q^{89} + q^{90} + q^{91}$

34:
(85645873456234584567345612345845673456123458345672345612345834567
2345612345834567234561234 58)

[1] q^{92}

Neighbours of a point in 0:

1: [56] $q + q^2 + 2q^3 + 3q^4 + 4q^5 + 5q^6 + 6q^7 + 6q^8 + 6q^9 + 6q^{10} + 5q^{11} + 4q^{12} + 3q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 1:

0: [1] 1

1: [15] $-1 + q + q^2 + 2q^3 + 3q^4 + 3q^5 + 3q^6 + 2q^7 + q^8$

2: [30] $q^5 + 2q^6 + 4q^7 + 5q^8 + 6q^9 + 5q^{10} + 4q^{11} + 2q^{12} + q^{13}$

3: [10] $q^{10} + q^{11} + 2q^{12} + 2q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 2:

1: [6] $1 + q + 2q^2 + q^3 + q^4$

2: [16] $-1 - q^2 + q^3 + 2q^4 + 4q^5 + 4q^6 + 4q^7 + 2q^8 + q^9$

3: [18] $q^6 + 2q^7 + 4q^8 + 4q^9 + 4q^{10} + 2q^{11} + q^{12}$

4: [3] $q^9 + q^{10} + q^{11}$

5: [12] $q^{10} + 2q^{11} + 3q^{12} + 3q^{13} + 2q^{14} + q^{15}$

8: [1] q^{16}

Neighbours of a point in 3:

1: [1] 1

2: [9] $q + 2q^2 + 3q^3 + 2q^4 + q^5$

3: [15] $-1 - q^2 - q^3 + q^4 + 3q^5 + 5q^6 + 5q^7 + 3q^8 + q^9$

5: [18] $q^7 + 3q^8 + 5q^9 + 5q^{10} + 3q^{11} + q^{12}$

6: [1] q^{10}

7: [3] $q^{11} + q^{12} + q^{13}$

9: [6] $q^{11} + 2q^{12} + 2q^{13} + q^{14}$

10: [3] $q^{14} + q^{15} + q^{16}$

Neighbours of a point in 4:

2: [15] $1 + q + 2q^2 + 2q^3 + 3q^4 + 2q^5 + 2q^6 + q^7 + q^8$

4: [6] $-1 - q^2 - q^4 + q^5 + q^6 + 2q^7 + q^8 + 2q^9 + q^{10} + q^{11}$

5: [20] $q^4 + q^5 + 2q^6 + 3q^7 + 3q^8 + 3q^9 + 3q^{10} + 2q^{11} + q^{12} + q^{13}$
8: [15] $q^8 + q^9 + 2q^{10} + 2q^{11} + 3q^{12} + 2q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 5:

2: [3] $1 + q + q^2$
3: [9] $q^2 + 2q^3 + 3q^4 + 2q^5 + q^6$
4: [1] q^3
5: [15] $-1 - q^2 - q^3 + 2q^5 + 4q^6 + 6q^7 + 4q^8 + 2q^9$
7: [3] $q^9 + q^{10} + q^{11}$
8: [3] $q^8 + q^9 + q^{10}$
9: [9] $q^8 + 2q^9 + 3q^{10} + 2q^{11} + q^{12}$
10: [9] $q^{10} + 2q^{11} + 3q^{12} + 2q^{13} + q^{14}$
11: [1] q^{13}
12: [3] $q^{14} + q^{15} + q^{16}$

Neighbours of a point in 6:

3: [20] $1 + q + 2q^2 + 3q^3 + 3q^4 + 3q^5 + 3q^6 + 2q^7 + q^8 + q^9$
6: [0] $-1 - q^2 - q^3 + q^7 + q^8 + q^{10}$
9: [30] $q^5 + 2q^6 + 3q^7 + 4q^8 + 5q^9 + 5q^{10} + 4q^{11} + 3q^{12} + 2q^{13} + q^{14}$
14: [6] $q^{11} + q^{12} + q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 7:

3: [6] $1 + q + 2q^2 + q^3 + q^4$
5: [12] $q^3 + 2q^4 + 3q^5 + 3q^6 + 2q^7 + q^8$
7: [7] $-1 - q^2 + q^5 + q^6 + 2q^7 + q^8 + 2q^9 + q^{10} + q^{11}$
10: [18] $q^6 + 2q^7 + 4q^8 + 4q^9 + 4q^{10} + 2q^{11} + q^{12}$
12: [12] $q^{10} + 2q^{11} + 3q^{12} + 3q^{13} + 2q^{14} + q^{15}$
17: [1] q^{16}

Neighbours of a point in 8:

2: [1] 1
4: [3] $q + q^2 + q^3$
5: [12] $q^2 + 2q^3 + 3q^4 + 3q^5 + 2q^6 + q^7$
8: [12] $-1 - q^2 - q^3 + q^5 + 3q^6 + 4q^7 + 4q^8 + 2q^9 + q^{10}$
10: [18] $q^7 + 2q^8 + 4q^9 + 4q^{10} + 4q^{11} + 2q^{12} + q^{13}$
13: [6] $q^{10} + q^{11} + 2q^{12} + q^{13} + q^{14}$
15: [4] $q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 9:

3: [4] $1 + q + q^2 + q^3$

5: [12] $q^2 + 2q^3 + 3q^4 + 3q^5 + 2q^6 + q^7$
6: [1] q^4
9: [13] $-1 - q^2 - q^3 - q^4 + q^5 + 3q^6 + 4q^7 + 4q^8 + 3q^9 + 2q^{10}$
10: [8] $q^7 + 2q^8 + 2q^9 + 2q^{10} + q^{11}$
11: [6] $q^9 + q^{10} + 2q^{11} + q^{12} + q^{13}$
12: [6] $q^{10} + q^{11} + 2q^{12} + q^{13} + q^{14}$
14: [2] $q^{11} + q^{12}$
16: [4] $q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 10:
3: [1] 1
5: [6] $q + 2q^2 + 2q^3 + q^4$
7: [3] $q^3 + q^4 + q^5$
8: [3] $q^4 + q^5 + q^6$
9: [4] $q^4 + 2q^5 + q^6$
10: [14] $-1 - q^2 - q^3 - q^4 + 3q^6 + 6q^7 + 5q^8 + 3q^9 + q^{10}$
12: [12] $q^8 + 3q^9 + 4q^{10} + 3q^{11} + q^{12}$
13: [1] q^{10}
14: [1] q^{11}
15: [6] $q^{11} + 2q^{12} + 2q^{13} + q^{14}$
16: [2] $q^{12} + q^{13}$
18: [3] $q^{14} + q^{15} + q^{16}$

Neighbours of a point in 11:
5: [4] $1 + q + q^2 + q^3$
9: [18] $q^2 + 2q^3 + 4q^4 + 4q^5 + 4q^6 + 2q^7 + q^8$
11: [12] $-1 - q^2 - q^3 - q^4 + 3q^7 + 3q^8 + 4q^9 + 3q^{10} + 2q^{11} + q^{12}$
12: [6] $q^6 + q^7 + 2q^8 + q^9 + q^{10}$
16: [12] $q^9 + 2q^{10} + 3q^{11} + 3q^{12} + 2q^{13} + q^{14}$
19: [1] q^{13}
22: [3] $q^{14} + q^{15} + q^{16}$

Neighbours of a point in 12:
5: [2] $1 + q$
7: [2] $q^2 + q^3$
9: [3] $q^2 + q^3 + q^4$
10: [12] $q^3 + 3q^4 + 4q^5 + 3q^6 + q^7$
11: [1] q^5

12: [13] $-1 - q^2 - q^3 - q^4 - q^5 + 2q^6 + 5q^7 + 6q^8 + 4q^9 + q^{10}$
15: [6] $q^9 + 2q^{10} + 2q^{11} + q^{12}$
16: [6] $q^9 + 2q^{10} + 2q^{11} + q^{12}$
17: [1] q^{10}
18: [6] $q^{11} + 2q^{12} + 2q^{13} + q^{14}$
21: [1] q^{13}
23: [3] $q^{14} + q^{15} + q^{16}$

Neighbours of a point in 13:
8: [10] $1 + q + 2q^2 + 2q^3 + 2q^4 + q^5 + q^6$
10: [10] $q^3 + q^4 + 2q^5 + 2q^6 + 2q^7 + q^8 + q^9$
13: [10] $-1 - q^2 - q^3 + q^5 + 2q^6 + 3q^7 + 3q^8 + 2q^9 + 2q^{10}$
15: [20] $q^7 + 2q^8 + 3q^9 + 4q^{10} + 4q^{11} + 3q^{12} + 2q^{13} + q^{14}$
20: [1] q^{11}
24: [5] $q^{12} + q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 14:
6: [1] 1
9: [10] $q + 2q^2 + 2q^3 + 2q^4 + 2q^5 + q^6$
10: [10] $q^4 + q^5 + 2q^6 + 2q^7 + 2q^8 + q^9 + q^{10}$
14: [5] $-1 - q^2 + q^5 + q^6 + 2q^7 + q^8 + q^9 + q^{11}$
16: [20] $q^6 + 2q^7 + 3q^8 + 4q^9 + 4q^{10} + 3q^{11} + 2q^{12} + q^{13}$
18: [10] $q^{10} + q^{11} + 2q^{12} + 2q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 15:
8: [1] 1
10: [9] $q + 2q^2 + 3q^3 + 2q^4 + q^5$
12: [9] $q^4 + 2q^5 + 3q^6 + 2q^7 + q^8$
13: [3] $q^4 + q^5 + q^6$
15: [13] $-1 - q^2 - q^3 - q^4 + q^6 + 4q^7 + 4q^8 + 4q^9 + 3q^{10} + q^{11}$
18: [9] $q^8 + 2q^9 + 3q^{10} + 2q^{11} + q^{12}$
21: [3] $q^{11} + q^{12} + q^{13}$
23: [3] $q^{12} + q^{13} + q^{14}$
24: [3] $q^{11} + q^{12} + q^{13}$
25: [3] $q^{14} + q^{15} + q^{16}$

Neighbours of a point in 16:
9: [3] $1 + q + q^2$
10: [3] $q^2 + q^3 + q^4$

11: [3] $q^3 + q^4 + q^5$

12: [9] $q^4 + 2q^5 + 3q^6 + 2q^7 + q^8$

14: [3] $q^3 + q^4 + q^5$

16: [13] $-1 - q^2 - q^3 - q^4 + 2q^6 + 4q^7 + 4q^8 + 4q^9 + 2q^{10} + q^{11}$

18: [9] $q^8 + 2q^9 + 3q^{10} + 2q^{11} + q^{12}$

22: [3] $q^{10} + q^{11} + q^{12}$

23: [9] $q^{11} + 2q^{12} + 3q^{13} + 2q^{14} + q^{15}$

27: [1] q^{16}

Neighbours of a point in 17:

7: [4] $1 + q + q^2 + q^3$

12: [24] $q^2 + 2q^3 + 4q^4 + 5q^5 + 5q^6 + 4q^7 + 2q^8 + q^9$

17: [0] $-1 - q^2 - q^3 - q^4 - q^5 + q^7 + 2q^8 + q^9 + q^{10}$

18: [24] $q^7 + 2q^8 + 4q^9 + 5q^{10} + 5q^{11} + 4q^{12} + 2q^{13} + q^{14}$

26: [4] $q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 18:

10: [3] $1 + q + q^2$

12: [6] $q^2 + 2q^3 + 2q^4 + q^5$

14: [1] q^3

15: [6] $q^4 + 2q^5 + 2q^6 + q^7$

16: [6] $q^4 + 2q^5 + 2q^6 + q^7$

17: [1] q^6

18: [13] $-1 - q^2 - q^3 - q^4 - q^5 + 4q^7 + 6q^8 + 5q^9 + 3q^{10}$

23: [12] $q^9 + 3q^{10} + 4q^{11} + 3q^{12} + q^{13}$

24: [1] q^{11}

25: [3] $q^{12} + q^{13} + q^{14}$

26: [2] $q^{13} + q^{14}$

28: [2] $q^{15} + q^{16}$

Neighbours of a point in 19:

11: [35] $1 + q + 2q^2 + 3q^3 + 4q^4 + 4q^5 + 5q^6 + 4q^7 + 4q^8 + 3q^9 + 2q^{10} + q^{11} + q^{12}$

19: [0] $-1 - q^2 - q^3 - q^4 - q^6 + q^7 + q^9 + q^{10} + q^{11} + q^{13}$

22: [21] $q^6 + q^7 + 2q^8 + 2q^9 + 3q^{10} + 3q^{11} + 3q^{12} + 2q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 20:

13: [21] $1 + q + 2q^2 + 2q^3 + 3q^4 + 3q^5 + 3q^6 + 2q^7 + 2q^8 + q^9 + q^{10}$

20: [0] $-1 - q^2 - q^4 + q^7 + q^9 + q^{11}$

24: [35] $q^4 + q^5 + 2q^6 + 3q^7 + 4q^8 + 4q^9 + 5q^{10} + 4q^{11} + 4q^{12} + 3q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 21:

12: [10] $1 + q + 2q^2 + 2q^3 + 2q^4 + q^5 + q^6$
15: [20] $q^3 + 2q^4 + 3q^5 + 4q^6 + 4q^7 + 3q^8 + 2q^9 + q^{10}$
21: [5] $-1 - q^2 - q^3 - q^4 - q^6 + q^7 + q^8 + 2q^9 + 2q^{10} + 2q^{11} + q^{12} + q^{13}$
23: [10] $q^6 + q^7 + 2q^8 + 2q^9 + 2q^{10} + q^{11} + q^{12}$
25: [10] $q^{10} + 2q^{11} + 2q^{12} + 2q^{13} + 2q^{14} + q^{15}$
29: [1] q^{16}

Neighbours of a point in 22:

11: [5] $1 + q + q^2 + q^3 + q^4$
16: [20] $q^2 + 2q^3 + 3q^4 + 4q^5 + 4q^6 + 3q^7 + 2q^8 + q^9$
19: [1] q^5
22: [10] $-1 - q^2 - q^3 - q^4 - q^5 + q^6 + 2q^7 + 3q^8 + 3q^9 + 3q^{10} + 2q^{11} + q^{12}$
23: [10] $q^7 + q^8 + 2q^9 + 2q^{10} + 2q^{11} + q^{12} + q^{13}$
27: [10] $q^{10} + q^{11} + 2q^{12} + 2q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 23:

12: [3] $1 + q + q^2$
15: [2] $q^3 + q^4$
16: [6] $q^2 + 2q^3 + 2q^4 + q^5$
18: [12] $q^4 + 3q^5 + 4q^6 + 3q^7 + q^8$
21: [1] q^5
22: [1] q^6
23: [14] $-1 - q^2 - q^3 - q^4 - q^5 + 3q^7 + 5q^8 + 6q^9 + 4q^{10} + q^{11}$
25: [4] $q^{10} + 2q^{11} + q^{12}$
26: [3] $q^{11} + q^{12} + q^{13}$
27: [3] $q^{10} + q^{11} + q^{12}$
28: [6] $q^{12} + 2q^{13} + 2q^{14} + q^{15}$
30: [1] q^{16}

Neighbours of a point in 24:

13: [3] $1 + q + q^2$
15: [12] $q^2 + 2q^3 + 3q^4 + 3q^5 + 2q^6 + q^7$
18: [6] $q^6 + q^7 + 2q^8 + q^9 + q^{10}$
20: [1] q^3
24: [12] $-1 - q^2 - q^3 + q^5 + 2q^6 + 4q^7 + 3q^8 + 3q^9 + q^{10} + q^{11}$

25: [18] $q^8 + 2q^9 + 4q^{10} + 4q^{11} + 4q^{12} + 2q^{13} + q^{14}$
28: [4] $q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 25:

15: [4] $1 + q + q^2 + q^3$
18: [6] $q^2 + q^3 + 2q^4 + q^5 + q^6$
21: [2] $q^4 + q^5$
23: [8] $q^5 + 2q^6 + 2q^7 + 2q^8 + q^9$
24: [6] $q^3 + q^4 + 2q^5 + q^6 + q^7$
25: [13] $-1 - q^2 - q^3 - q^4 - q^5 + q^6 + 3q^7 + 4q^8 + 4q^9 + 4q^{10} + 2q^{11}$
28: [12] $q^9 + 2q^{10} + 3q^{11} + 3q^{12} + 2q^{13} + q^{14}$
29: [1] q^{12}
30: [4] $q^{13} + q^{14} + q^{15} + q^{16}$

Neighbours of a point in 26:

17: [1] 1
18: [12] $q + 2q^2 + 3q^3 + 3q^4 + 2q^5 + q^6$
23: [18] $q^4 + 2q^5 + 4q^6 + 4q^7 + 4q^8 + 2q^9 + q^{10}$
26: [7] $-1 - q^2 - q^3 - q^4 + 2q^7 + q^8 + 2q^9 + 2q^{10} + 2q^{11} + q^{12} + q^{13}$
28: [12] $q^8 + 2q^9 + 3q^{10} + 3q^{11} + 2q^{12} + q^{13}$
30: [6] $q^{12} + q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 27:

16: [4] $1 + q + q^2 + q^3$
22: [6] $q^2 + q^3 + 2q^4 + q^5 + q^6$
23: [18] $q^3 + 2q^4 + 4q^5 + 4q^6 + 4q^7 + 2q^8 + q^9$
27: [12] $-1 - q^2 - q^3 - q^4 - q^5 + 2q^7 + 4q^8 + 4q^9 + 4q^{10} + 2q^{11} + q^{12}$
28: [12] $q^9 + 2q^{10} + 3q^{11} + 3q^{12} + 2q^{13} + q^{14}$
31: [3] $q^{13} + q^{14} + q^{15}$
32: [1] q^{16}

Neighbours of a point in 28:

18: [3] $1 + q + q^2$
23: [9] $q^2 + 2q^3 + 3q^4 + 2q^5 + q^6$
24: [1] q^3
25: [9] $q^4 + 2q^5 + 3q^6 + 2q^7 + q^8$
26: [3] $q^5 + q^6 + q^7$
27: [3] $q^6 + q^7 + q^8$
28: [15] $-1 - q^2 - q^3 - q^4 - q^5 - q^6 + 2q^7 + 4q^8 + 6q^9 + 5q^{10} + 3q^{11} + q^{12}$

30: [9] $q^{10} + 2q^{11} + 3q^{12} + 2q^{13} + q^{14}$
31: [1] q^{13}
32: [3] $q^{14} + q^{15} + q^{16}$

Neighbours of a point in 29:
21: [6] $1 + q + q^2 + q^3 + q^4 + q^5$
25: [30] $q^2 + 2q^3 + 3q^4 + 4q^5 + 5q^6 + 5q^7 + 4q^8 + 3q^9 + 2q^{10} + q^{11}$
29: [0] $-1 - q^2 - q^3 - q^4 - q^5 + q^8 + q^9 + q^{10} + q^{11} + q^{12}$
30: [20] $q^7 + q^8 + 2q^9 + 3q^{10} + 3q^{11} + 3q^{12} + 3q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 30:
23: [3] $1 + q + q^2$
25: [6] $q^2 + 2q^3 + 2q^4 + q^5$
26: [3] $q^3 + q^4 + q^5$
28: [18] $q^4 + 3q^5 + 5q^6 + 5q^7 + 3q^8 + q^9$
29: [1] q^6
30: [15] $-1 - q^2 - q^3 - q^4 - q^5 - q^6 + q^7 + 3q^8 + 5q^9 + 6q^{10} + 4q^{11} + 2q^{12}$
32: [9] $q^{11} + 2q^{12} + 3q^{13} + 2q^{14} + q^{15}$
33: [1] q^{16}

Neighbours of a point in 31:
27: [15] $1 + q + 2q^2 + 2q^3 + 3q^4 + 2q^5 + 2q^6 + q^7 + q^8$
28: [20] $q^3 + q^4 + 2q^5 + 3q^6 + 3q^7 + 3q^8 + 3q^9 + 2q^{10} + q^{11} + q^{12}$
31: [6] $-1 - q^2 - q^3 - q^4 + 2q^7 + q^8 + 2q^9 + 2q^{10} + 2q^{11} + q^{13}$
32: [15] $q^8 + q^9 + 2q^{10} + 2q^{11} + 3q^{12} + 2q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 32:
27: [1] 1
28: [12] $q + 2q^2 + 3q^3 + 3q^4 + 2q^5 + q^6$
30: [18] $q^4 + 2q^5 + 4q^6 + 4q^7 + 4q^8 + 2q^9 + q^{10}$
31: [3] $q^5 + q^6 + q^7$
32: [16] $-1 - q^2 - q^3 - q^4 - q^5 - q^6 + q^7 + 2q^8 + 4q^9 + 5q^{10} + 5q^{11} + 3q^{12} + 2q^{13}$
33: [6] $q^{12} + q^{13} + 2q^{14} + q^{15} + q^{16}$

Neighbours of a point in 33:
30: [10] $1 + q + 2q^2 + 2q^3 + 2q^4 + q^5 + q^6$
32: [30] $q^3 + 2q^4 + 4q^5 + 5q^6 + 6q^7 + 5q^8 + 4q^9 + 2q^{10} + q^{11}$
33: [15] $-1 - q^2 - q^3 - q^4 - q^5 - q^6 + q^8 + 2q^9 + 4q^{10} + 4q^{11} + 4q^{12} + 3q^{13} + 2q^{14} + q^{15}$
34: [1] q^{16}

Neighbours of a point in 34:

33: [56] $1 + q + 2q^2 + 3q^3 + 4q^4 + 5q^5 + 6q^6 + 6q^7 + 6q^8 + 6q^9 + 5q^{10} + 4q^{11} + 3q^{12} + 2q^{13} + q^{14} + q^{15}$

34: [0] $-1 - q^2 - q^3 - q^4 - q^5 - q^6 + q^{10} + q^{11} + q^{12} + q^{13} + q^{14} + q^{16}$

Annals of Discrete Mathematics 26 (1985) 49–58

Performance of Subset Generating Algorithms

Margaret Carkeet and Peter Eades

Department of Computer Science
University of Queensland
St. Lucia, Queensland
AUSTRALIA

Abstract

This note reports on some tests of several algorithms for generating the subsets of fixed size of a set. In particular, the speed of execution is compared.

1. Introduction

In this note the results of tests of algorithms for generating all subsets of size k of a set of size n (sometimes called *combinations)* are reported. We are concerned with testing the *speed* of the algorithms. No complexity analysis is applied; we are merely reporting the results of some tests.

There are eight such algorithms known to the authors.

1 *BER:* From [1]. We tested the optimized version of the algorithm, described in [9] (page 186).

2 *CHASE:* From [3].

3 *EMK:* From [5]. An optimized version (from B. D. McKay, private communication) was tested.

4 *EE:* Even's version (in [7], page 42) of Ehrlich's algorithm in [6].

5 *LS:* The optimized (third) version from [8].

6 *LEX:* The usual lexicographic algorithm. It is described in all standard texts, including [9] (page 181).

7 *RD:* The "revolving door" algorithm presented in [10] (subroutine NXSRD on page 30).

8 *EHR:* The very strong minimal change algorithm described in [4] and [2]. Note that this algorithm works only for restricted values of n and k. For this reason, and because it is much slower than the others, this algorithm was not tested.

Some of these algorithms have "minimal change" properties, that is, successively generated subsets differ from each other by a small amount. To describe these properties we need to consider the data structures used to represent subsets. The elements of the sets are represented by the integers $1,2,\ldots,n$. A k-subset S of an n-set can be represented as a *bitvector* $(b_1, b_2, \ldots, b_n)$, where b_x is 1 if x is in S and 0 if x is not in S. Alternatively, if $S=\{s_1, s_2, \ldots, s_k\}$ where $s_1 < s_2 < \ldots < s_k$, then S can be represented by the *ordered array* $(s_1, s_2, \ldots, s_k)$.

(Aside : All the algorithms above can be implemented using either data structure. For testing each algorithm was implemented using the data structure which made it faster: bitvectors were used for BER and EE, all the others used ordered arrays. It is usually easy to convert an ordered array algorithm to a bitvector algorithm without effecting performance significantly. The reverse conversion, however, often reduces performance.)

The minimal change properties are:

1 *WMCP* (Weak Minimal Change Property): Successively generated bitvectors differ in at most two positions. This means that the next subset is formed from the previous one by deleting one element and adding another. This property holds for all the above algorithms except LEX.

2 *SMCP* (Strong Minimal Change Property): Successively generated ordered arrays differ in only one position. Note that this implies WMCP. This property holds for EHR, CHASE, EMK, and EE.

3 *VSMCP* (Very Strong Minimal Change Property): Successively generated bitvectors differ in two adjacent positions. This implies SMCP. It holds for EHR only.

These properties are discussed in detail in [5].

2. The Results

The first seven algorithms above were tested on a Perkin-Elmer 3220 running UNIX. These language used was Pascal, and the programs were run under two different systems: the Berkeley Pascal to p-code compiler, and a UQ Pascal to C compiler.

The Berkeley system reports the number of statements executed, and this was used as an indication of running time. The UNIX *time* utility was used to give an indication of the execution time under the UQ system. The two different Pascal systems and the two different timing systems were in substantial agreement, and only the results from the Berkeley system are quoted here.

The authors recognize the dangers of this type of measurement. The *time* utility is a little sensitive to the machine load at the time of execution. It is quite probable that a different programmer, a different language, a different hardware configuration, could have produced different results. Every effort was made to minimize the effect of these differences, but we admit that at best, only the first few digits of our results are significant. To obtain more significance a full complexity analysis (along the lines of the analysis of LEX in [9]) would be required.

With the exception of LEX and RD, all the algorithms tested are fast in the sense that the average time to generate a subset is bounded by a constant, independent of n and k. Further, these algorithms are *loopless*, or *uniformly bounded*, which roughly means that the time to generate each subset is constant, independent of n and k. (See [9] for a precise definitions of these properties.) LEX and RD do not have these properties when k is close to n.

The graph in figure 1 summarizes the results. The tables from which figure 1 was derived are in figure 2. The vertical axis in figure 1 is the average number of Pascal statements executed per subset produced. The average was taken over $n=5$ to $n=12$. The horizontal axis represents the range of k; the leftmost value is $k=2$, and the rightmost is $k=n-2$. The other value of k are dispersed linearly between the left and rightmost.

Some statement counts for larger values of n are given in figure 3.

3. Conclusions

All the algorithms except EHR are reasonably simple and can be coded in a few pages. LEX is very simple and takes only a few minutes to write.

No algorithm (except EHR) uses more than $O(n)$ space; this is insignificant in comparison to time requirements.

The main result of the tests is that LS is significantly faster than any of the others. An implementation of LS on a VAX11/750 generates a subset about every 45 microseconds; on a Cyber 172/2 it takes about one third of this time.

In an application, each subset has to be processed in some way. If the processing time dominates the generation time, then the processing time also determines the size of the largest problem that can be tackled. However, if the processing time is about the same or less than the generation time, then the generation time imposes a limit on the largest problem which can be tackled: for instance, in an hour of CPU time on the Cyber172/2, LS can process every 15-subset of a 30-set. Hand optimized assembler, or a supercomputer, could improve this limit, but not significantly.

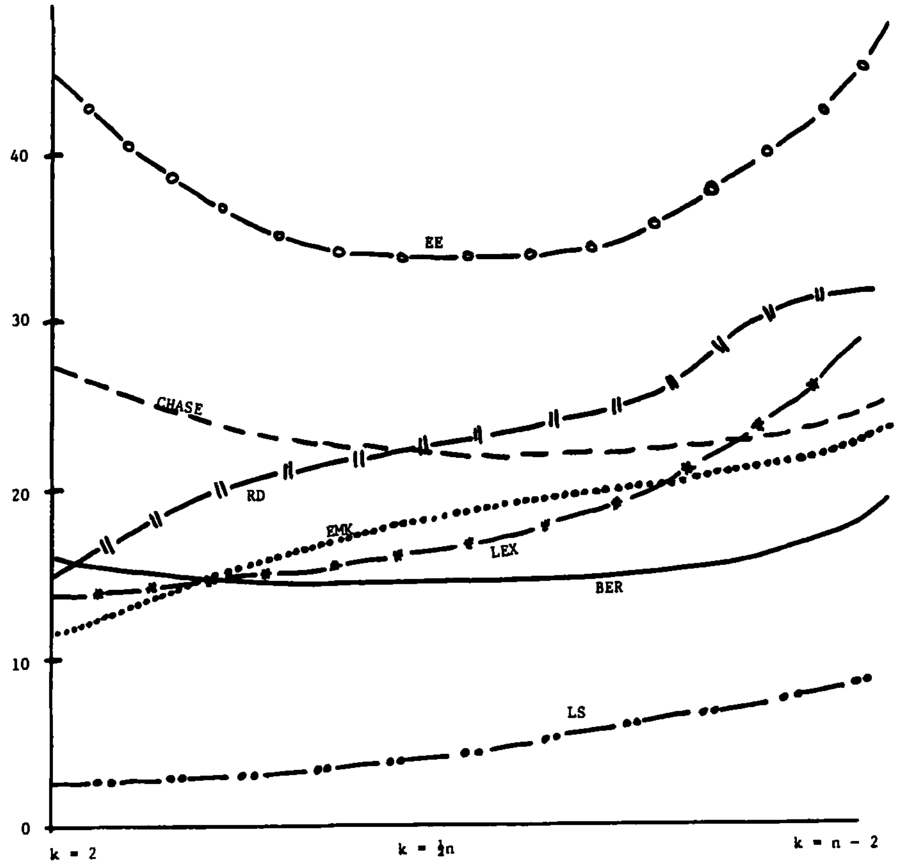


Figure 1

The only disadvantage of using LS is that it does not have SMCP. EMK, about 4 times slower than LS, is the fastest algorithm with this property. If the processing is significantly faster with SMCP, then EMK should be used. Also, if the processing time dominates generation time, then a minor speedup from SMCP may justify EMK.

The problem of finding a fast algorithm which has VSMCP is open.

Finally we note that LEX is surprisingly fast. The simplicity of this algorithm (it requires no clever stack implementation), makes it attractive.

Figure 2a. BER - number of statements executed

$n=5$	173	161							
$n=6$	252	296	244						
$n=7$	347	495	539	319					
$n=8$	458	770	1058	794	434				
$n=9$	585	1133	1893	1733	1245	533			
$n=10$	728	1596	3152	3408	3080	1680	680		
$n=11$	887	2171	4959	6179	6771	4523	2403	803	
$n=12$	1062	2870	7454	10506	13574	10734	7166	4902	1602
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$

Figure 2b. CHASE - number of statements executed

$n=5$	246	240							
$n=6$	371	459	358						
$n=7$	527	797	789	493					
$n=8$	716	1285	1554	1237	664				
$n=9$	940	1956	2803	2740	1861	848			
$n=10$	1201	2854	4719	5486	4557	2646	1076		
$n=11$	1501	3989	7520	10142	9995	7134	3670	1313	
$n=12$	1842	5427	11461	17593	20085	17054	10748	4902	1602
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$

Figure 2c. EMK - number of statements executed									
$n=5$	189	187							
$n=6$	260	364	319						
$n=7$	334	610	712	428					
$n=8$	411	928	1353	1122	603				
$n=9$	491	1321	2316	2451	1750	755			
$n=10$	574	1792	3678	4733	4232	2481	973		
$n=11$	660	2344	5519	8361	9012	6677	3475	1168	
$n=12$	749	2980	7922	13806	17452	15625	10185	4613	1429
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$

Figure 2d. EE - number of statements executed									
$n=5$	325	322							
$n=6$	515	653	516						
$n=7$	760	1173	1181	752					
$n=8$	1067	1938	2367	1929	1065				
$n=9$	1440	3007	4316	4288	3003	1423			
$n=10$	1886	4448	7334	8594	7300	4409	1877		
$n=11$	2411	6334	11793	15916	15903	11689	6293	2381	
$n=12$	3021	8744	18138	27695	31828	27569	17989	9646	3001
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$

Figure 2e. LS - number of statements executed									
$n=5$	37	57							
$n=6$	48	90	98						
$n=7$	61	134	182	147					
$n=8$	76	191	310	319	208				
$n=9$	93	263	495	619	515	277			
$n=10$	112	352	752	1104	1122	776	358		
$n=11$	133	460	1098	1846	2214	1882	1116	447	
$n=12$	156	589	1552	2934	4048	4080	2980	1541	548
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$

Figure 2f. LEX - number of statements executed									
$n=5$	152	177							
$n=6$	222	333	293						
$n=7$	306	558	628	446					
$n=8$	402	866	1188	1076	642				
$n=9$	511	1270	2056	2266	1720	886			
$n=10$	633	1783	3328	4324	3988	2604	1183		
$n=11$	768	2418	5113	7654	8314	6598	3793	1538	
$n=12$	916	3188	7533	12769	15970	14914	10393	5333	1956
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$

Figure 2g. RD - number of statements executed									
$n=5$	174	261							
$n=6$	251	514	338						
$n=7$	342	888	736	621					
$n=8$	445	1404	1295	1592	713				
$n=9$	561	2086	2406	3474	1974	1160			
$n=10$	690	2957	3873	6786	4641	3658	1260		
$n=11$	832	4040	5913	12213	9726	9655	4221	1903	
$n=12$	987	5358	8656	20629	18697	22466	11836	7109	2003
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$

Figure 3. Number of Pascal statements executed.

n	k	BER	CHASE	EMK	EE	LS	LEX	RD
14	3	4687	9350	4515	15481	917	5184	8790
14	6	44419	66540	53237	105357	11433	49046	57163
14	9	27239	43211	40520	72131	11518	41038	59380
14	12	1339	2249	1970	4484	777	3000	2965
18	3	10339	22664	8727	39387	1915	11370	19488
18	6	278111	432264	292287	>500000	58801	284169	326006
18	9	>500000	>500000	>500000	>500000	214523	>500000	>500000
18	12	267183	403018	379161	>500000	111599	400449	450742
18	15	11343	19151	17366	38858	6483	25905	30451

References

[1] J. R. Bitner, G. Ehrlich, and E. M. Reingold, "Efficient Generation of the Binary Reflected Gray Code and its Applications", *Communications of the Association for Computing Machinery,* 19 (1978) 517-521.

[2] Margaret Carkeet and Peter Eades, "An Implementation of a Minimal Change Algorithm", Technical Report No. 45, Department of Computer Science, University of Queensland, January 1983.

[3] Philip J. Chase, "Algorithm 382: Combinations of M out of N objects", *Communications of the Association for Computing Machinery,* 13 (1970) 368.

[4] Peter Eades, Michael Hickey and Ronald C. Read, "Some Hamilton Paths and a Minimal Change Algorithm", *Journal of the Association for Computing Machinery* 31 (1984) 19-29.

[5] Peter Eades and Brendan McKay, "An Algorithm for Generating Subsets of a Fixed Size with a Strong Minimal Change Property", *Information Processing Letters* 19 (1984) 131-133.

[6] Gideon Ehrlich, "Loopless Algorithms for Generating Permutations, Combinations, and other Combinatorial Configurations", *Journal of the Association for Computing Machinery,* 20 (1973) 500-513.

[7] Shimon Even, *Algorithmic Combinatorics,* Macmillan (1973).

[8] Clement W. H. Lam and Leonard H. Soicher, "Three New Combination Algorithms with the Minimal Change Property", *Communications of the Association for Computing Machinery,* 25 (1982) 555-559.

[9] Edward M. Reingold, Jurg Nievergelt and Narsingh Deo, *Combinatorial Algorithms, Theory and Practice,* Prentice Hall (1977).

[10] Albert Nijenhuis and Herbert S. Wilf, *Combinatorial Algorithms,* Monographs in Computer Science and Applied Mathematics, Academic Press, (1975).

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 59-66

The Computational Complexity of Finding Subdesigns in Combinatorial Designs

Charles J. Colbourn and Marlene J. Colbourn

Department of Computer Science
University of Waterloo
Waterloo, Ontario
CANADA

Douglas R. Stinson

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
CANADA

Abstract

Algorithms for determining the existence of subdesigns in a combinatorial design are examined. When $\lambda=1$, the existence of a subdesign of order d in a design of order v can be determined in $O(v^{\log d})$ time. The order of the smallest subdesign can be computed in polynomial time. In addition, determining whether a design has a subdesign of maximal possible order (a "head") requires polynomial time. When $\lambda>1$, the problems are apparently significantly more difficult: we show that deciding whether a block design has *any* non-trivial subdesign is NP-complete.

1. Introduction

A *(balanced incomplete) block design* of *order* v, denoted $B[k,\lambda;v]$, is a v-set V of elements together with a collection B of k-element subsets of V called *blocks*, with $v>k$; each 2-subset of V appears in precisely λ of the blocks. A *Steiner system* is a block design with $\lambda=1$; a *Steiner triple system* is a Steiner system with $k=3$. A *subdesign* of a $B[k,\lambda;v]$ (V,B) is a $B[k,\lambda';v']$ (V',B') such that $V' \subseteq V$ and $B' \subseteq B$; subdesigns are non-trivial when $v' > k$, and either $v > v'$ or $\lambda > \lambda'$. A design without non-trivial subdesigns is called *simple*. It is easy to see that $v \geq (k-1)v'+1$; when equality is met, the subdesign is called a *head* [5].

We examine the computational complexity of determining the existence of various types of subdesigns. A primary motivation is that the numbers and types of subdesigns are often used as invariants in distinguishing isomorphism classes of designs (see, for example, [4]). We show that for Steiner systems, most problems involving subdesigns are computationally straightforward. Determining simplicity requires polynomial time, as does locating subdesigns of any fixed constant size. When the order of the desired subdesign d increases as a function of the order of the design v, the complexity of this algorithm becomes superpolynomial, but still subexponential. At the other extreme, we show that polynomial time suffices to determine whether a Steiner system has a head.

For block designs in general, however, finding subdesigns is not an easy task. In fact, we show that deciding whether a B[3,3;v] design has a subdesign is NP-complete, and hence is unlikely to have any efficient solution. This result shows that deciding the existence of a subdesign of specified size is also NP-complete.

2. Subdesigns of Steiner systems

Doyen and Wilson [7] showed that given two admissible orders v and w ($v>w$), there is a Steiner triple system of order v having a subdesign of order w if and only if $v \geq 2w+1$; thus, there are many possible orders for subdesigns. It is also known that there is a simple Steiner triple system for every admissible order [6]. We are concerned with the related question of determining when a particular Steiner system has subdesigns; the following lemma is straightforward:

Lemma 2.1: Simplicity of Steiner systems can be decided in polynomial time.

Proof:

A subdesign has the property that every block intersects the subdesign in 0, 1, or k elements. Therefore, given a subset S of elements to be placed in a subdesign, we can *close* this set, by repeatedly introducing all elements of blocks intersecting the set in more than one element. When this closure procedure introduces no new elements, the set obtained forms a subdesign. Taking any single element and closing, one obtains a trivial subdesign of order 1. Taking any pair of elements and closing yields a block, another trivial subdesign. Taking any three elements not appearing in a block together, and closing, yields either a proper subdesign or the design itself.

Simplicity of Steiner systems can therefore be easily tested by applying closure to each set of three elements in turn. The design is simple if and only if the subdesigns obtained are trivial in each case. ○

Lemma 2.1 gives a method for determining whether there are any proper subdesigns; it is worth noting that the method can easily be modified to find the smallest subdesign. One simply retains the minimum size of a nontrivial subdesign encountered. Of course, this method does not help us determine whether there is a subdesign of specified size, or the size of the largest subdesign. Nonetheless, a similar closure method will answer these questions in subexponential time.

Lemma 2.2: Determining the presence of a subdesign of order d in a Steiner system of order v can be accomplished in $v^{O(\log d)}$ time.

Proof:

Every subdesign of order d is generated by a set of log d elements, in the following sense: given these log d elements, closure produces the subdesign. This can be easily seen by induction. Thus it suffices to enumerate all sets of log d elements chosen from the v elements in the design. Closure is applied to each; the design has a subdesign of order d if and only if one of these closures produces one. Since closure can be applied in polynomial time to each set of $\log d$ elements, and there are $v^{O(\log d)}$ such sets, the total time required is $v^{O(\log d)}$. ●

It is worth remarking that when d is a constant, the time bound in lemma 2.2 is polynomial. Lemma 2.2 also gives a subexponential time algorithm for finding the largest subdesign; in practice, the subexponential method operates quite quickly, since its worst case is realized only when there is a significant number of subdesigns (such as the projective and affine spaces). In many of the worst cases, the design has a subdesign of maximal order, a head. Although we are unable to determine the size of the maximal subdesign in polynomial time, we can make one step in this direction, by determining whether the design has a head.

Lemma 2.3: The existence of heads in Steiner systems can be decided in polynomial time.

Proof:

The key observation here is that every block intersects a head in 1 or k elements. The algorithm for finding a head operates as follows. At any given stage, we mark an element as "in" the head, "out" of the head, or "undecided". The usual closure operation enables us to mark all elements of a block "in" when two elements of the block are marked "in" already. In searching for heads, given a block containing an element marked "in" and an element marked "out", all other elements can be marked "out".

The algorithm proceeds by usual backtracking. Initially, an element is chosen to be marked "out". At a general step, a block is chosen involving an element which is "out" and all other elements unmarked. One of these k-1 elements must be marked "in" and the remainder "out". It should be noted that it is possible for closure to produce a contradiction, i.e. a specification of an element as both "in" and "out"; in this event, the elements chosen to be "in" cannot form a head, and we simply backtrack. If no contradiction arises, once one is marked "in", the two closure operations will increase the number of elements marked "in" by a factor of k-1 (at least), since closure produces a subdesign. This ensures that the depth of the backtrack is $O(\log_{k-1} v)$. Since at each level of the backtrack there is a fixed number k-1 of choices, the backtrack operates in time $(k\text{-}1)^{O(\log_{k-1} v)}$, which is a polynomial in v. ●

These lemmas establish that

Theorem 2.4: In polynomial time, one can decide whether a Steiner system has a subdesign, find the order of the smallest subdesign, determine the existence of subdesigns of fixed constant order, and determine the existence of a head. In subexponential time, one can determine the existence of a subdesign of specified order and the order of the maximal subdesign.

3. Subdesigns of Block Designs

The results from section 2 all generalize in the obvious manner if we are to determine subdesigns with the *same* λ as that of a given block design. In this section, however, we show that the situation is dramatically different when, as in our definition, subdesigns are allowed to have smaller λ. Here we establish that even deciding whether a design has a nontrivial subdesign is NP-complete, even for B[3,3;v] designs.

This NP-completeness result is predicated on the use of a combinatorial structure called a "Latin background", which has been used previously in establishing numerous NP-completeness results for design-theoretic problems [1,2]. Given an n-vertex r-regular graph G, a *Latin background* for G, denoted LB[G;m,s], $s \geq n$, is an s by s symmetric array with elements chosen from $\{1,2,...,m\}$. Each diagonal position contains the element m. In the first n rows, each entry is either empty, or contains an element from $\{r+1,...,m\}$; in the latter s-n rows, each position contains an element from $\{1,...,m\}$. Every element appears at most once in each row and in each column (hence $m \geq s$). Finally, the pattern of empty positions is precisely an adjacency matrix for the graph G -- hence the term "background". We require the following result from [1]:

Lemma 3.1: Let G be an n-vertex r-regular graph. A Latin background LB[G;m,m] exists for every even $m \geq 2n$. Moreover, such a background can be constructed in time which is polynomial in m. ●

The Latin backgrounds formed in lemma 3.1 are partial symmetric Latin squares which can be completed if and only if G is r-edge-colourable. Edge-colouring graphs is NP-complete [9,12], and hence so is completing symmetric Latin squares [1]. This underlies the following result:

Theorem 3.2: Determining whether a B[3,3;v] design has a subdesign is NP-complete.

Proof:

Membership in NP is straightforward; hence, we need only reduce a known NP-complete problem to our problem. We reduce the problem of determining whether a cubic graph is 3-edge-colourable [9]. Given an arbitrary n-vertex cubic graph G, we construct in polynomial time a B[3,3;6s-3] design which has a subdesign if and only if G is 3-edge-colourable. First, we construct a LB[G;2s,2s], where $s \geq n$ is the smallest integer for which $2s-1$ is a prime. It is important to note that s is O(n) [8]. In the Latin background, we then eliminate all occurrences of the last element, 2s, leaving the diagonal empty. The entries of the last row (and column) are moved into the corresponding diagonal positions, after which the last row and column are deleted. Rows and columns are then simultaneously interchanged so that position (i,i) contains i; that is, the 2s-1 by 2s-1 square is idempotent. Denote this modified square by IB. We will also employ a 2s-1 by 2s-1 idempotent symmetric Latin square SL having no subsquares. For example, one could take the square whose (i,j) entry is $i+j$ (mod $2s-1$), and interchange rows and columns to make it idempotent; this has no subsquares since $2s-1$ was chosen to be prime.

The B[3,3;6s-3] we create has elements $\{x_1, \ldots, x_{2s-1}\}$, $\{y_1, \ldots, y_{2s-1}\}$, and $\{z_1, \ldots, z_{2s-1}\}$; it contains the following blocks:

1. $\{\{x_i,y_i,z_i\} \mid 1 \leq i \leq 2s-1\}$, each three times.
2. $\{\{y_i,y_j,z_k\} \mid 1 \leq i < j \leq 2s-1$, SL contains k in position $(i,j)\}$, each three times.
3. $\{\{z_i,z_j,x_k\} \mid 1 \leq i < j \leq 2s-1$, SL contains k in position $(i,j)\}$, each three times.
4. $\{\{x_i,x_j,y_k\} \mid 1 \leq i < j \leq 2s-1$, position (i,j) of IB is nonempty and contains $k\}$, each three times.

5. $\{\{x_i,x_j,y_k\} \mid 1 \le i < j \le 2s-1$, position (i,j) of IB is empty, $1 \le k \le 3\}$.

All blocks are repeated, except those arising from empty positions in IB. A nontrivial subdesign of this $B[3,3;6s-3]$ design must involve all $6s-3$ elements, since any subdesign induces a subsquare on the $\{y_i\}$ and on the $\{z_i\}$ and SL has no nontrivial subsquares. Then the only possible nontrivial subdesign is a $B[3,1;6s-3]$, i.e. a *decomposition* of the design into designs with smaller λ. Any $B[3,1;6s-3]$ induces a symmetric Latin square on the $\{x_i\}$ which is a completion of IB, and conversely. Hence the $B[3,3;6s-3]$ has a decomposition (and hence a nontrivial subdesign) if and only if IB is completable, which holds if and only if the original graph is 3-edge-colourable. ●

Theorem 3.2 strongly suggests that algorithms for subdesign problems applied to block designs in general will have exponential running time in the worst case.

4. Future Research

A very general formulation of algorithmic questions about subdesigns could ask when a block design $B[k,\lambda;v]$ contains a $B[k',\lambda';v']$. Of course, $v \ge v'$, $k \ge k'$, and $\lambda \ge \lambda'$. In this paper, we have considered only the case $k=k'$. When $k=k'$ and $v=v'$, this is the question of decomposability of designs, studied in [2,11]. Another question of this type arises when one takes $v=v'$ and $k>k'$; this is the question of when a design contains a *nested* design (see, for example, [3,10,13,14]). The complexity of determining whether a design has a nested design remains open.

Acknowledgements

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada and Simon Fraser University for supporting the Workshop on Latin Squares held at Simon Fraser, July-August 1983, during which this research was done. Research of the authors is funded by NSERC Canada under grants numbered A5047 (CJC), A5483 (MJC), and U0217 (DRS).

References

[1] C.J. Colbourn, "Embedding partial Steiner triple systems is NP-complete", *Journal of Combinatorial Theory* A35 (1983) 100-105.

[2] C.J. Colbourn and M.J. Colbourn, "The complexity of decomposing block designs", *Annals of Discrete Mathematics*, to appear.

[3] C.J. Colbourn and M.J. Colbourn, "Nested triple systems", *Ars Combinatoria* 16 (1983) 27-34.

[4] M.J. Colbourn and R.A. Mathon, "On cyclic Steiner 2-designs", *Annals of Discrete Mathematics* 7 (1980) 215-253.

[5] F.N. Cole, L.D. Cummings, and H.S. White, "The complete enumeration of triad systems in 15 elements", *Proceedings of the National Academy of Sciences, U.S.A.* 3 (1917) 197-199.

[6] J. Doyen, "Sur la structure de certains systemes triples de Steiner", *Mathematische Zeitschrift* 111 (1969) 289-300.

[7] J. Doyen and R.M. Wilson, "Embeddings of Steiner triple systems", *Discrete Mathematics* 5 (1973) 229-239.

[8] D.R. Heath Brown and H. Iwaniec, "On the difference between consecutive primes", *Inventiones Mathematicae* 55 (1979) 49-69.

[9] I. Holyer, "The NP-completeness of edge-coloring", *SIAM Journal on Computing* 10 (1981) 718-720.

[10] M. Jimbo and S. Kuriki, "Constructions of nested designs", *Ars Combinatoria* 16 (1983) 275-285.

[11] E.S. Kramer, "Indecomposable triple systems", *Discrete Mathematics* 8 (1974) 173-180.

[12] D. Leven and Z. Galil, "NP-completeness of finding the chromatic index of regular graphs", *Journal of Algorithms* 4 (1983) 35-44.

[13] C.C. Lindner and D.R. Stinson, "On the spectra for the conjugate invariant subgroups of perpendicular arrays", *Ars Combinatoria* 18 (1984) 51-60.

[14] E. Mendelsohn, "Perpendicular arrays of triple systems", *Annals of Discrete Mathematics* 7 (1980) 89-94.

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 67–136

Algorithmic Aspects of Combinatorial Designs: A Survey

Marlene Jones Colbourn

Department of Computer Science
University of Waterloo
Waterloo, Ontario
CANADA N2L 3G1

Abstract

We present a survey of existing results concerning algorithmic aspects of combinatorial design theory. The scope within design theory includes block designs and restricted families thereof, Latin squares and their variants, pairwise balanced designs, projective planes and related geometries, and Hadamard and related matrices. However, the emphasis is on t-designs, particularly balanced incomplete block designs and Steiner systems. There are many different algorithmic aspects of combinatorial design theory which could be discussed here; we focus upon isomorphism testing and invariants, colouring, nesting, resolvability, decomposing, embedding and completing, orienting and directing, as well as algorithmic aspects of intersection graphs. Also included is a brief discussion of some general algorithmic techniques including backtracking, hill-climbing, greedy and orderly algorithms.

1. Introduction

Research on combinatorial design theory extends from the mid-eighteen hundreds to the present. Throughout the intervening decades, researchers have examined many interesting problems in combinatorial design theory. Some of the questions and solutions proposed are algorithmic in nature.

It is our intent here to examine some of the algorithmic aspects and issues in combinatorial design theory. Within design theory, we include block designs and variations thereof such as balanced incomplete block designs, pairwise balanced designs and Steiner systems, Latin squares and their variants, projective planes and related geometries, and Hadamard and related matrices. Over the years, researchers have examined a wide variety of aspects concerning block designs and related combinatorial configurations, many of which are algorithmic in nature, have algorithmic solutions, or exploit algorithmic tools. We discuss some of these aspects and issues. First we present some necessary definitions, as well as some of the essential background regarding computational complexity. This is followed by an introduction to common algorithmic techniques such as backtracking, branch-and-bound, hill-climbing, orderly and greedy algorithms.

We cannot hope to provide a complete survey of all algorithmic aspects of combinatorial design theory. Rather we focus on particular problems: isomorphism testing and invariants, colouring blocks and elements, nesting, resolvability, decomposing, embedding and completing, orienting and directing, as well as some algorithmic aspects of intersection graphs. While presenting results in these areas, we try to provide the reader with examples of different types of proofs. Hence, our choice of which proofs to present is influenced by our desire to provide representative proofs without encumbering the reader with excessive detail.

One aspect of combinatorial design theory which we do not survey here is existence, despite the fact that many proofs of existence include direct or recursive constructions which are algorithmic in nature. To survey this area would be an enormous task which is beyond the scope of this paper.

1.1 Definitions

1.1.1 Design Theory Definitions

For a general introduction to combinatorial design theory, the reader should consult [S10]. A *t-design* t-B$[k,\lambda;v]$ is a pair (V,B) where B is a collection of k-subsets called *blocks* of the v-set V, such that every t-subset of V is contained in precisely λ blocks of B. $|V| = v$ is referred to as the *order* of the design. Some researchers refer to λ as the *balance factor* or *index*. From these parameters, one can calculate the *replication factor* r, the number of blocks to which each element belongs, as $\lambda\left(\frac{(v-1)}{(t-1)}\right)/\left(\frac{(k-1)}{(t-1)}\right)$. The total number of blocks in the design, b, is then vr/k. A *balanced incomplete block design* (BIBD), denoted B$[k,\lambda;v]$, is a t-design with $t=2$. A BIBD is said to be *symmetric* if $v=b$. Symmetric designs with $\lambda=1$ are *projective planes*, and when $\lambda=2$, they are referred to as *biplanes*.

Early research concerning t-designs was initiated by the investigation of a restricted class of designs, Steiner systems. A *Steiner system*, denoted S(t,k,v) is a t-design with $\lambda=1$; e.g. a t-B$[k,1;v]$ design. Two families of Steiner systems which have received an enormous amount of attention are *Steiner triple systems*, which are S$(2,3,v)$ designs and *Steiner quadruple systems*, denoted S$(3,4,v)$.

Twofold triple systems, B[3,2;v] designs, have also been the focus of much research. In particular, researchers have examined various directing or orderings of the blocks to form Mendelsohn triple systems and directed triple systems. A ***Mendelsohn triple system*** is a B[3,2;v] design in which the blocks are cyclic 3-tuples, such that each ordered pair of elements occurs in exactly one block. For example, the block (x,y,z) contains the pairs (x,y), (y,z) and (z,x), but not the pairs (y,x), (z,y), (x,z). On the other hand, a ***directed triple system*** is a B[3,2;v] design in which the blocks are ordered 3-tuples, such that the block (x,y,z) contains the pairs (x,y), (x,z) and (y,z). Again, each ordered pair of elements must occur in exactly one block. These definitions can be extended to higher values of k.

A ***pairwise balanced design*** (PBD) is a generalization of a BIBD, in which the blocks may be of different sizes. If $K=\{k_1,...,k_m\}$ is a set of positive integers, a PBD B[K,λ;v] is a pair (V,B); B is a collection of blocks from a v-set V of elements such that every pair of elements appears in exactly λ blocks of B and every block of B has cardinality belonging to the set K. A ***partially balanced incomplete block design*** (PBIBD) is another generalization of a BIBD. In this case, each pair of elements need not appear the same number of times. If $\Lambda=\{\lambda_1,...,\lambda_m\}$ is a set of positive integers, a PBIBD B[k,Λ;v] is an arrangement of v elements into k-subsets such that each pair occurs together in λ_i blocks for some $\lambda_i \in \Lambda$.

Two designs (V_1,B_1) and (V_2,B_2) are ***isomorphic*** if and only if there exists a bijection $f:V_1 \to V_2$ such that $b \in B_1$ if and only if $f(b) \in B_2$. An ***automorphism*** of a design is an isomorphism of the design with itself. The set of all automorphisms forms a group under the usual composition of mappings, called the ***automorphism group***.

A design of order v is ***cyclic***, denoted t-CB[k,λ;v], when its automorphism group contains a v-cycle. A $t-$CB[k,λ;v] design can be represented as a t-B[k,λ;v] design with elements $\{0,...,v-1\}$ for which if $\{a_1, \ldots, a_k\}$ is a block, $\{a_1+1,...,a_k+1\}$ (addition performed modulo v) is also a block. A cyclic design is always isomorphic to a design (V,B) for which $V=Z_v=\{0,1,...,v-1\}$ and the mapping $f:i \to i+1$ (mod v) is an automorphism.

The most common representation for cyclic designs is in terms of difference sets. A (v,k,λ) (cyclic) ***difference set*** $D=\{d_1,...,d_k\}$ is a collection of k residues modulo v such that for any residue $x \neq 0$ (mod v), the congruence $d_i - d_j \equiv x$ (mod v) has exactly λ solution pairs (d_i,d_j) with $d_i,d_j \in D$. Every (v,k,λ) difference set generates a cyclic symmetric BIBD, whose blocks are $B(i)=\{d_1+i,...,d_k+i\}$ (mod v), $i=0,...,v-1$. The difference set is often referred to as the ***starter*** or ***base block*** of the symmetric design.

A (v,k,λ) *difference family* is a collection of such sets $D_1,...,D_n$ each of cardinality k such that each residue $x \neq 0 \pmod v$ has exactly λ solution pairs (d_i,d_j) with $d_i,d_j \in D_m$ for some m. A difference family is said to be *planar* or *simple* if $\lambda=1$. Each (v,k,λ) difference family generates a cyclic BIBD in the same manner as before. For example, the difference family (0,1,4) (0,2,7) generates the cyclic S(2,3,13) design with $V = \{0,1,...,12\}$. This definition is really not sufficiently general; for example, an S(2,3,15) design cannot be represented as a difference family, as defined above. However, it is possible for the design to be generated by 2 starter blocks modulo 15, when one includes the 5 blocks generated by the *extra starter block* (0,5,10). We will call a S(2, k,v) design *cyclic* if the design can be generated by m starter blocks modulo v, possibly with the extra starter block $(0, m',2m',...,(k-1)m')$ where $b=mv+m'$, $m'<v$. The definition can be generalized for larger values of t and λ in the obvious manner.

Consider two difference sets, D_1 and D_2, having the same parameters. If $D_2=tD_1+s \pmod v$ for some integers t and s, D_1 and D_2 are *equivalent* difference sets. If $D_1=tD_1+s \pmod v$, t is a *multiplier* of D_1. The mappings $x \to tx+i \pmod v$, $i=0,...,v-1$, are isomorphisms of the associated symmetric block designs.

This idea can also be extended to difference families. Consider two (v,k,λ) difference families D and E; $D=\{D_1,\ldots,D_n\}$ and $E=\{E_1,...,E_n\}$. They are equivalent if for some integers t and $s_1,\ldots,s_n$, $\{E_1,\ldots,E_n\} = \{tD_1+s_1,\ldots,tD_n+s_n\} \pmod v$, t is a multiplier of the difference family D. The mappings $x \to tx+i \pmod v$, $i=0,...,v-1$, are isomorphisms of the associated block designs. The collection of multiplier automorphisms of a given difference set or family form a group under composition called the *multiplier group*. There do exist cyclic designs which possess different automorphism and multiplier groups.

Recall that a design is cyclic if it has an automorphism consisting of a single cycle of length v. We can define k-rotational designs in an analogous way; a design is *k-rotational*, $k \geq 1$, if it has an automorphism fixing one element and permuting the remaining elements in k cycles of length $(v-1)/k$ each. (Note that this k is not related to the block size k).

A *partial parallel class* (PPC) of a design D is a collection of mutually disjoint blocks of D. A *parallel class* (PC) is a PPC in which each element of V occurs exactly once; in other words, a PC contains v/k blocks. A design is said to be *resolvable* if the b blocks can be partitioned into disjoint parallel classes. In the case of STS, a resolvable STS is referred to as a *Kirkman triple system*. STS exist when $v \equiv 1,3 \pmod 6$ [K3, R5]; obviously, when $v \equiv 1 \pmod 6$, a STS cannot be resolvable. However, if after removing an element and the blocks in which that element appears, one can partition the design into parallel classes, the original STS is referred to as a *nearly Kirkman triple system*. Given a design, the largest PPC(s) contained therein is said to be *maximum*. A PPC is *maximal* if there is no block of the remaining design which is mutually disjoint with all of the blocks in the PPC; hence, the PPC cannot be extended.

A *Latin square* of order n is an $n \times n$ array; each entry is an element from the set $\{1,...,n\}$. Each row contains each element exactly once, and each column contains each element exactly once. Two Latin squares of order n, L_1 and L_2, are said to be *orthogonal* if, for any i, $1 \leq i \leq n$, the n positions which contain i in L_1 are occupied in L_2 by $1,2,...,n$, each occurring once.

Algebraically, a Latin square is the multiplication table of a quasigroup. A *quasigroup* is a pair $(A,*)$; A is a set of elements and $*$ is a binary operation such that for $a,b \in A$, the equations $a*x=b, a*b=y$, and $z*a=b$ have unique solutions for x,y and z. A quasigroup is *commutative* if $a*b=b*a$ for all $a,b \in A$. A quasigroup is *idempotent* if $a*a=a$ for all $a \in A$. The corresponding Latin square is *symmetric* when the quasigroup is commutative.

A *partial Latin square* of order n is an $n \times n$ array; each entry is either empty or else it contains an element from $\{1,...,n\}$. Each row (column) contains each element at most once. One important investigation of the structure of partial Latin squares aims to characterize partial Latin squares which can be *completed* to Latin squares without the addition of rows, columns, or elements.

A *Howell design* $H(n,2t)$, with $t \leq n \leq 2t-1$, is a square array of side n, where cells are either empty or contain an unordered pair of elements chosen from a set X of size $2t$ such that: (1) each member of X occurs exactly once in each row and column of the array, and (2) each pair of elements of X occurs in at most one cell of the array. A *Room square* of side n (n odd) is an $H(n,n+1)$ design. It follows that, in this case, each pair of elements of X occurs in exactly one cell of the array.

A *Hadamard matrix* of order n is an $n \times n$ (1,-1)-matrix which satisfies $HH^T = nI$, where H^T is the transpose of H. A Hadamard matrix is in *standard form* is all entries of the first row and column are 1. For a Hadamard matrix to exist n must be 1, 2 or $4m$, $m \geq 1$. A *Hadamard design* is a symmetric $B[2m-1,m-1;4m-1]$ design. Such designs exist if and only if an Hadamard matrix of side $4m$ exists. To see this, given an Hadamard matrix in standard form, remove column 1 and row 1. Then replace all -1's by 0's. The result is an incidence matrix of a Hadamard design [S10,p224]. Two Hadamard matrices are *Hadamard equivalent* if one can be obtained from the other by a finite series of the following operations: multiply a row by -1, multiply a column by -1, interchange any two rows, or interchange any two columns.

1.1.2 Graph Theory Definitions

A *hypergraph* is a pair (V,E) such that E is a subset of the powerset of V. A hypergraph is k*−uniform* if each edge of E is of cardinality k. A *graph* is a 2-uniform hypergraph. The *incidence graph* of a hypergraph (V,E) has a vertex for each member of V and for each member of E. Whenever a vertex v belongs to an edge e of E, the corresponding vertices in the incidence graph are adjacent.

A *strongly regular graph* has parameters n, k, p, q. It is a n-vertex graph regular of degree k satisfying the constraints that two adjacent vertices x, y have p common neighbours (for any x,y) and two non-adjacent vertices have q common neighbours (for any x,y). A strongly regular graph is, in fact, a 2-class association scheme. An *association scheme* consists of a set V together with a partition of the set of 2-element subsets of V into s classes R_i, $1 \leq i \leq s$, satisfying the following two conditions:
(1) given $p \in V$, the number v_i of $q \in V$ with $\{p,q\} \in R_i$ depends only on i;
(2) given $\{p,q\} \in R_k$, the number $p(i,j,k)$ of $r \in V$ with $\{p,r\} \in R_i$, $\{r,q\} \in R_j$ depends only on i,j,k.

A $1-$*factor* of a graph is a spanning subgraph which is regular of degree 1. A $1-$*factorization* of a graph is a collection of edge-disjoint 1-factors whose union is the entire graph.

For additional graph theory definitions, the reader should consult [B11].

1.2 Computational Complexity

Throughout this paper, we describe various algorithmic solutions to some of the interesting problems in combinatorial design theory. For many of these problems, efficient algorithms have been developed. By *efficient*, we mean algorithms which require at most a polynomial amount of time -- polynomial in the size of the input on a conventional computing device or a unit-cost RAM (random-access machine) [A1]. We employ the standard "O" notation to denote an upper bound on an algorithm's running time. Saying a function $f(n)$ is $O(g(n))$ means that $|f(n)| \leq c|g(n)|$ for some constant c and for all $n \geq 0$. For example, saying an algorithm is "order n^2" or $O(n^2)$ implies that the running time of the algorithm is bounded by the function cn^2 for some constant c and for all values of n. A *polynomial time* algorithm is defined to be one whose time complexity function is $O(p(n))$ for some polynomial p. In the case $p(n)=n$, the algorithm is said to be *linear*. If an algorithm is $O(n^{\log n})$, it is *subexponential*. An *exponential* algorithm is an $O(x^n)$ algorithm for some $x \geq 2$.

Polynomial time algorithms are much more desirable than those requiring exponential time. For convincing evidence of this, see [G4, p7], where the running time of various algorithms is compared. For example, consider an $O(n^2)$ and an $O(2^n)$ algorithm. On input of size 30, they would require .0009 seconds and 17.9 minutes of execution time, respectively. When the input size is increased to 60, the time requirements have increased to .0036 seconds and 366 centuries, respectively.

The distinction between polynomial and exponential-time algorithms was first made in [C3, E1]. More importantly, Edmonds [E1] equated polynomial time algorithms with the notion of "good" or efficient algorithms. The class P is defined to be the set of all problems which have polynomial time algorithms. A problem is considered ***intractable*** if it is so hard that no polynomial time algorithm can possibly solve it [G4].

The earliest intractability results are the undecidability results of Turing. He proved, for example, that it is impossible to specify any algorithm which, given an arbitrary computer program and an arbitrary input to that program, can decide whether or not the program will eventually halt when applied to that input [T4]. Other problems have since been shown to be undecidable; see [G4, L5, H11] for a discussion.

The first examples of intractable decidable problems were obtained in the early sixties [H10]; for a discussion of these problems, see [G4, C39]. Unlike these early examples, most of the apparently intractable problems encountered in practice are decidable and can be solved in nondeterministic polynomial time. However, this means that none of the proof techniques developed so far is powerful enough to verify the apparent intractability of these problems.

The class NP consists of all problems that can be solved in polynomial time on a nondeterministic Turing machine; NP stands for ***nondeterministic polynomial***. One can think of these problems as being solvable in polynomial time if one can guess the correct computational path to follow. In 1971, Cook [C38] proved that a particular problem in NP, 3-CNF-Satisfiability, has the property that every other problem in NP can be polynomially reduced to it. If this satisfiability problem is solved with a polynomial time algorithm, so can every problem in NP; if any problem in NP is intractable, the satisfiability problem is also intractable. Hence, in some sense, the satisfiability problem is the "hardest" problem in NP. A wide variety of problems have now been shown to be of equivalent difficulty to the satisfiability problem; for example, see [G4, K1]. This equivalence class of the "hardest" problems in NP is the class of ***NP−complete*** problems.

The question of whether or not the NP-complete problems are intractable is one of the major open questions of computer science. If a problem is shown to be NP-complete, this is generally accepted as strong evidence that the problem is difficult and that it is highly unlikely that a polynomial time algorithm will be developed to solve the problem.

To establish that a problem R is NP-complete, one must first show that the problem is in NP and that some other NP-complete problem Q is polynomial-time reducible to R. A problem Q is *polynomial−time reducible* to a problem R if the required transformation can be executed by a polynomial time deterministic algorithm. If this is the case, a polynomial time algorithm to solve problem R will also provide a polynomial algorithm for problem Q. Examples of polynomial-time reductions and NP-completeness proofs are provided later in this paper.

Any problem, whether a member of NP or not, to which we can transform an NP-complete problem will have the property that it cannot be solved in polynomial time unless P=NP. Such a problem is said to be *NP−hard*, since it is at least as hard as the NP-complete problems; see [G4] for an excellent discussion of both NP-complete and NP-hard problems.

2. General Algorithmic Techniques

There are several common algorithmic approaches which researchers have employed when searching for or generating combinatorial configurations with particular properties. The most notable of these are orderly algorithms, greedy algorithms, hill-climbing, backtracking, and branch and bound algorithms. These techniques are by no means restricted to use within combinatorics, but rather are common approaches employed within many different mathematical applications. We briefly describe each of these methods here and mention some of the uses of each approach within combinatorial design theory. Again, we cannot hope to survey all of the relevant literature, but rather cite representative examples of each technique's applicability.

Probably the most common of the aforementioned algorithmic techniques is backtracking, which is a method of implicitly searching all possible solutions in a systematic manner. A formal definition of the backtrack search technique can be found in [B5]. More recent expositions of the method can be found in [A1, H14, P1].

Backtrack programming is a method for the systematic enumeration of a set of vectors. Therefore, it is applicable to discrete problems in which possible solutions can be described by vectors, the elements of which are members of a particular finite set. The vectors need not all have the same dimension. The first task in employing a backtrack algorithm is to establish a one-to-one correspondence between the combinatorial configurations and the vectors or sequences. For a BIBD, the vector could represent the blocks of the design in lexicographically increasing order. In order to employ a backtrack, there must be some notion of lexicographical ordering, since a backtracking algorithm typically enumerates the vectors starting from the lexicographically smallest vector.

A backtrack algorithm is best described by explaining its operation in the midst of the backtrack process. We include here a presentation based on [P1]. Suppose that a complete vector $(x_1,x_2,...,x_r)$ has just been constructed. At this point, the vector may be made available to some other routine for processing; for example, at this point, one would check to see whether the generated vector satisfies the particular constraints or properties for which one is searching. Upon return to the backtrack procedure, an attempt is made to find a new r^{th} element. This new element is selected from the set X_r of elements which can occur in the r^{th} position, given the values of the elements that are in the first $r-1$ positions of the vector. If X_r is not empty, its first member may be selected, deleted from the set X_r, and inserted into the vector in the r^{th} slot. We may now have another complete vector or we may have to select further elements in the vector; regardless, the set X_r has been reduced by one member. If, however, X_r was empty, it is necessary to backtrack to the previous component of the vector and replace element x_{r-1}. Clearly, x_{r-1} can only be replaced if the set of remaining possible members for that element, X_{r-1}, is not empty. If X_{r-1} is non-empty, we choose a new element, delete it from X_{r-1}, replace element x_{r-1}, and move forward again. We now must form a new set X_r of elements which are now possible candidates for the r^{th} slot in the vector. Of course, if X_{r-1} was empty, it would have been necessary to backtrack even further.

In this way, the vector is built up, one element at a time. Whenever one runs out of possible candidates for the current slot in the vector, one backtracks. If one wants the search to be exhaustive, the backtracking process continues until all possible candidates for the first vector position have been examined. Often, however, one simply wants to find a solution, in which case the backtrack is terminated when the first solution is encountered.

Ideally, each X_k, $1 \le k \le r$, should be easy to compute and contain as few elements as possible. In order to reduce the portion of the solution space which is being searched, one wants to determine at an early stage in the construction of the partial vector that it is not suitable or whether it has already been examined in some other form. This usually entails exploiting information concerning the automorphisms of the current, and possibly previous, partial solutions.

The backtrack method aims at doing all of its validity testing according to the problem specifications during the formation of the vectors. At the other extreme, one could enumerate all complete vectors and only test the complete vectors for validity. Of course, one need not settle for either extreme, but rather incorporate some testing while forming the vectors and leave the rest until a completed vector is obtained. Obviously, the benefit of doing the extra work during the production of the vectors will only be felt if there is a substantial reduction in the number of vectors produced. However, from experience, it appears that when generating combinatorial configurations, where there tend to be many partial solutions which correspond to almost-completed vectors, it is crucial to eliminate unsuitable partial solutions; hence, the extra work during the production stage seems critical.

A special variation of backtrack for optimization problems is *branch* and *bound*. In a backtrack algorithm, a partial vector (and all its descendents e.g. larger vectors which include this particular partial vector) are excluded from the search if the partial vector already violates the constraints. We can associate high costs with such infeasible vectors and zero cost with those vectors which do satisfy the constraints. Then a backtrack algorithm can be viewed as searching for a minimum cost vector. If one can associate a cost with each partial vector such that $cost(x_1,\ldots,x_{k-1}) \leq cost(x_1,\ldots,x_{k-1},x_k)$, for all possible values of k, one can view the generation problem as searching a tree of possible solutions in which the cost of a parent node is always less than or equal to the cost of its children. In such a case, if we have found a solution node S_1 with cost C, we would not examine the children of a partial solution node S_2 whose cost exceeds C, since all the children of S_2 will be of higher cost than C. This is the central idea in branch and bound. We do not *branch* from a node whose cost is higher than the cost of the minimum cost solution found so far. Of course, the *bound* is updated if a better solution is found. Therefore, in contrast to backtracking, a branch and bound algorithm extends the most promising partial solution, rather than the most recent. For more detailed descriptions of various branch and bound algorithms, see [A1, H14, P1].

Branch and bound techniques, as well as the more general version of backtracking, are common approaches to generating combinatorial configurations. For examples of the use of backtracking for generating designs and related configurations, see [C7, D2, G11, G5, I1, K8]. The trick to a successful backtrack is to prune the search by employing appropriate isomorphism rejection techniques. When generating designs, one can employ a backtracking algorithm either on a block-by-block basis or element-by-element. The former is the more common approach.

Unlike backtracking algorithms and variations thereof, ***hill−climbing*** is not exhaustive. Because of this, an algorithm based strictly on a hill-climbing method may not yield the optimal solution, but rather one which is only locally the best. For the same reason, this technique does not guarantee a solution.

Given an initial configuration or vector X and an evaluation function f, the basic hill-climbing algorithm moves to a new configuration X' if $f(X')<f(X)$. The algorithm halts when no further improvement can be made.

This search method has been employed to generate SBIBD [S2], mutually orthogonal Latin squares [T2], strong starters, and hence Room squares and Howell designs [D8], and STS [S5]. In order to employ a hill-climbing algorithm, there must be some sense of when one partial solution is better than another. In other words, an evaluation or cost function is required as in branch and bound algorithms. In the case of constructing BIBD, the evaluation function may be simply the number of element pairs which do not appear in the partial design. One also needs at least one technique for moving from one partial design to another. Ideally one wants to move to a better partial solution, but often hill-climbing algorithms are implemented such that one may move to a configuration of the same worth; in doing this, one must be careful to avoid cycling. However, one never moves to a configuration of less worth, as is the case in backtracking algorithms. In some cases, hill-climbing algorithms are implemented in conjunction with some backtracking, so that if a local optimum is reached that is unsuitable, the algorithm either backtracks or jumps to another location in the search space. In their search for strong starters, Dinitz and Stinson [D8] include very limited backtracking.

To have some hope of success with a hill-climbing algorithm, one needs a good evaluation function which is easy to compute and several fast methods of moving from one partial configuration to another. As an example of a successful hill-climbing algorithm which includes several appropriate heuristics for converting partial configurations, we present Dinitz and Stinson's research concerning generating strong starters [D8].

A ***strong starter*** of order n in an additive Abelian group G of odd order $n=2t+1$ is a set $S=\{\{x_1,y_1\},...,\{x_t,y_t\}\}$ which satisfies the following properties:

(i) $\{x_1,...,x_t,y_1,...,y_t\}=G-\{0\}$,
(ii) $\{\pm(y_i-x_i) \mid \{x_i,y_i\}\in S\}=G-\{0\}$,
(iii) $x_i+y_i\neq x_j+y_j$ if $i\neq j$, and $x_i+y_i\neq 0$, for any i.

Dinitz and Stinson use hill-climbing to find strong starters of order $n=2t+1$ in the cyclic group Z_n. To do this, we first need the notion of a ***partial strong starter***, which is a set $S'=\{\{x_1,y_1\},\ldots,\{x_r,y_r\}\}$, $1\leq r\leq t$, satisfying the following conditions:

(i) the x_i's and y_i's are distinct nonzero elements of Z_n;
(ii) $y_i - x_i \neq \pm(y_j - x_j)$ if $i \neq j$;
(iii) $x_i + y_i \neq x_j + y_j$ if $i \neq j$, and $x_i + y_i \neq 0$ if $0 \leq i \leq r$.

The *deficiency* of S' is $def(S') = t - r$; in other words, it is the number of "missing pairs". A partial strong starter S' is *maximal* if there exists no $\{u,v\} \subseteq Z_n$ such that $S' \cup \{\{u,v\}\}$ is a partial strong starter.

Consider a set of *differences* $D = \{1,2,...t\}$; D is a set of natural numbers. Without loss of generality, we can assume that $y_i > x_i$, $1 \leq i \leq r$; then $y_i - x_i = d_i \in D$, if $1 \leq i \leq r$. If an element $z \in Z_n - \{0\}$ is $\in \{x_i, y_i\}$ for some such set in S', z is said to be *used*; otherwise z is *unused*. Similarly, one can refer to a difference as being used or unused. Finally, $e \in Z_n - \{0\}$ is said to be a used or unused sum depending on whether or not $e = x_i + y_i$ for some i, $1 \leq i \leq r$.

A *state* of the hill-climbing algorithm is a partial strong starter S' together with two distinct unused elements u_1 and u_2, and an unused difference $d \in D$. Given a state of the algorithm, let $T_i = \{u_i - d, u_i + d\}, i = 1,2$, and let $T = T_1 \cup T_2$. The following operations can be performed on a state:

(i) matching u_i with an unused element. If there exists $w \in T_i$ such that w is an unused element and $u_i + w$ is an unused sum (for the appropriate $i =$ 1 or 2), let $S'' = S' \cup \{\{u_i, w\}\}$. If $def(S'') \neq 0$, choose a new u_1, u_2, d.
(ii) switching a pair. If $w \in T_i$ is a used element and $u_i + w$ is an unused sum, let $S'' = S' - \{\{x_j, y_j\}\} \cup \{\{w, u_i\}\}$, where $w = x_j$ or y_j, for some j, $1 \leq j \leq r$. Set $d = d_j$, $u_1 = u_{3-i}$, and $u_2 = y_j$, if $w = x_j$; if $w = y_j$, set $u_2 = x_j$.
(iii) backtracking. Revert to the previous state of the algorithm if (ii) or (iii) was the last operation performed.
(iv) switching a difference. Replace d by some other unused difference d'. Leave u_1, u_2 unchanged.
(v) switching a pair. Suppose $u_i - u_{3-i} = d_1 \in D$ is a used difference, and $u_1 + u_2$ is an unused sum. Then set $S'' = S' - \{\{x_{d_1}, y_{d_1}\}\} \cup \{u_1, u_2\}$; set $u_1 = x_{d_1}$, $u_2 = y_{d_1}$, and leave d unchanged.

The algorithm can now be described in terms of operations (i)-(v):

(1) Initialization: Set $def = t, S = \emptyset$, choose any distinct $u_1, u_2 \in Z_n - \{0\}$, $d \in D$.
(2) If operation (i) can be performed, do so and go to (8).
(3) If operation (ii) can be performed, do so and go to (2).

(4) If operation (iii) can be performed, do so and go to (3).
(5) If operation (iv) can be performed, do so and go to (2).
(6) If operation (v) can be performed, do so and go to (2).
(7) Stop; algorithm fails.
(8) Set $def = def - 1$, choose any distinct unused u_1, u_2 and d.
If $def \neq 0$ go to (2).
(9) Stop; algorithm succeeds.

It is important to note that no operation increases the deficiency and operation (i) decreases it by 1. There may be more than one way to perform an operation (ii) on a given state; one is selected at random. If a state is reached again, this time by backtracking, the first way to perform operation (ii) is excluded and one of the remaining ways is chosen at random.

In [D8] Dinitz and Stinson also present a probabilistic proof that the algorithm should run and succeed in $O(n^2)$ time. In fact, this hill-climbing algorithm has been successfully employed to generate strong starters, Room squares and Howell designs. A similar hill-climbing algorithms for STS, due to Stinson [S5], is also based on the notion of "switching", analogous to the switching heuristic employed in the strong starter case.

Anderson [A3] recently extended Dinitz and Stinson's hill-climbing approach to construct houses. Let n be a positive integer, S be a set of elements of size $2n$, and F be a partition of S into unordered pairs. A ***house*** of order n is a $2n \times 2n$ array H such that

(i) every cell of H is either empty or contains an unordered pair of distinct symbols of S,
(ii) every symbol occurs in precisely one cell of each row and each column of H,
(iii) the pairs in F each occur in precisely two cells of H, whereas every other pair of symbols occurs in exactly one cell of H,
(iv) the pairs in the first and second rows of H are precisely those in F,
(v) every column of H contains one pair from F.

The success of hill-climbing algorithms may in part be due to the richness of the solution space. If there are many solutions, one's chances of successfully climbing to a solution via relatively weak heuristics is better than in a sparse solution space. However, hill-climbing algorithms have not been employed sufficiently often for researchers to characterize problem spaces which will lend themselves well to the technique.

Greedy algorithms have the same flavour as hill-climbing algorithms in that they are concerned with local optimums. At any individual stage, a *greedy* algorithm selects that option which is "locally optimal" in some particular sense. For example, when colouring the elements of a design, one's decision criterion may concern the number of colours being used. Hence, a locally optimal partial solution is the one which employs the fewest colours. Of course, it may be impossible to extend this partial solution to a proper colouring of the given design, let alone an optimal colouring. Greedy algorithms for colouring STS are discussed in section 4 of this paper and in [C15].

One's decision criterion with regard to which element or object to select next may be very simple. For example, when constructing a spanning tree of a connected graph, one need only check that the edge being added does not create a cycle. This simple greedy algorithm always produces a spanning tree.

In general, it appears that when generating combinatorial configurations, greedy algorithms do not suffice. For example, consider the construction of an $n \times n$ latin square by filling in the entries one by one, checking at each stage that no entries in that row or column have been filled with the same symbol. There are examples in which this greedy algorithm will fail. One interesting question is to determine the smallest integer k such that a 'failed' partial $n \times n$ latin square can always be partitioned into k pieces, each of which can be extended into a $n \times n$ latin square; k is the *intricacy* of the problem. For the latin square problem, it has been determined that the intricacy is always between 2 and 4 [O1]. Other construction problems and their intricacy are also examined in [O1]; such results indicate when a greedy algorithm will succeed and can also be employed to suggest when such an algorithmic approach can be expected to suffice on average.

Although greedy algorithms have not been applied extensively in design theory, one problem which appears conducive to this type of approach is the construction of partial parallel classes (PPC). For example, to establish a lower bound on the size of a maximum PPC in a STS, Brouwer [B17] employs a greedy-style procedure which includes an exchange process when the current PPC one is constructing cannot be extended directly. Brouwer's bound is presented in section 5.

In the generation or search methods discussed so far -- backtrack, branch and bound, hill-climbing and greedy algorithms -- a particular solution may, in fact, be encountered more than once unless one incorporates an appropriate isomorphism rejection mechanism into the algorithm. This is usually done by exploiting automorphism information of the partial solutions. For some problems, an *orderly* algorithm is possible in which the combinatorial configurations are generated in canonical form, hence removing the problem of checking for duplicate solutions [R3].

We present the strategy here in terms of graphs, employing the corresponding vector notation of the upper triangle of the adjacency matrix. A graph can be represented by possibly many adjacency matrices. Thus, each graph may have many vector representations. To make the representation of a graph G unique, we define the *canonical form* of G to be the largest vector which is a vector representation of G.

Let (p,q)-graph denote a graph with p vertices and q edges. Typically in graph generation, one is given a list $L(p,q)$ of all nonisomorphic graphs with p vertices and q edges, and required to produce the list $L(p,q+1)$. In an orderly algorithm, the canonical form of every $(p,q+1)$-graph is obtained by changing some 0 to a 1 in the canonical form of some (p,q)-graph. If the 0 changed is required to be to the right of the rightmost 1 in the canonical form then the canonical form of each $(p,q+1)$-graph is produced from the canonical form of exactly one (p,q)-graph. This change from 0 to 1 is called an *augmentation*.

This allows one to start with an ordered list $L(p,q)$ of the canonical forms of the nonisomorphic (p,q)-graphs, and perform augmentation in every possible way on each member of $L(p,q)$. The resulting set of vectors contains canonical forms and other vector representations. However, since each required canonical form appears on the list exactly once, we simply test each graph to see whether it is canonical, and include it in $L(p,q+1)$ if and only if it is. Observe that, we determine whether a given vector representation is to be added to $L(p,q+1)$ without referring to what has already been added.

Orderly algorithms for graphs have been studied by a variety of researchers; for example, see [C30, R3]. Their application need not be restricted to graphs. Unfortunately, for many combinatorial problems, it appears to be difficult to generate the canonical form of one combinatorial configuration from the canonical form of a smaller one.

Elsewhere in this volume, Ivanov [I1] employs a combination of an orderly algorithm with traditional backtracking techniques to generate BIBD. The algorithm is orderly in the sense that one is generating canonical incidence matrices of the designs. In general, one is backtracking through the search tree (or solution space). However, not all branches of the search tree need be examined since it can be shown that they cannot contain canonical matrices of the desired designs; hence, the canonicity information is being employed to prune the search tree.

Orderly algorithms have also been employed to construct SQS [C19, C28, P7]; Phelps's algorithm [P7] is discussed in section 5.

Without techniques such as orderly algorithms, one is forced to incorporate isomorphism rejection into exhaustive generation methods such as backtrack or branch and bound algorithms, if one wants all possible solutions. We examine algorithms for isomorphism testing and the use of isomorphism invariants in section 3.

3. Isomorphism Testing and Invariants

3.1 Isomorphism Testing

The problem of deciding whether two graphs are isomorphic has attracted a significant amount of attention [C5]. One of the reasons is that although the problem is not known to be NP-complete, no algorithm to solve it in polynomial (or even subexponential) time is known [R4]. Over the years, many proofs have appeared demonstrating that testing isomorphism of random graphs can be done efficiently, and with high probability of success [B2, K2, L6]. It is of interest, therefore, to identify the difficult instances of the problem.

Corneil [C40] observed that practical isomorphism algorithms have the most difficulty with strongly regular graphs and other graphs obtained from combinatorial configurations. In a compilation of graphs which are hard for isomorphism algorithms, Mathon [M2] included solely graphs derived from combinatorial configurations.

To show that a subclass of graphs is difficult, one must at least establish that an algorithm to solve isomorphism in the subclass is powerful enough to solve graph isomorphism. Formally, one must show that deciding isomorphism of graphs in the subclass is polynomial time equivalent to graph isomorphism or *isomorphism complete*. For a survey of results concerning isomorphism completeness, see [B13]. Since that survey, however, other problems have been shown to be isomorphism complete. In particular, it it now known that

Theorem 3.1 [C34]: Testing isomorphism of block designs is isomorphism complete.

Theorem 3.2 [F1]: Testing isomorphism of 4-class association schemes is isomorphism complete.

Hence, it is unlikely that we will devise an efficient (polynomial-time) algorithm for block design isomorphism. Consequently, one is motivated to search for better algorithms for specific subcases.

Using a result of Tarjan, Miller [M12] showed that quasigroup isomorphism can be decided in $O(v^{\log v})$ time; the standard representation of an STS as a Steiner quasigroup yields a subexponential algorithm for deciding isomorphism in this case. Implementations of this algorithm are discussed in [C32, S6].

Although no polynomial-time algorithm is known for testing isomorphism of STS, Stinson found that in practice Miller's algorithm appears to run in time $O(v^4 \log v)$ [S4, S6]. Miller's algorithm can be easily extended to handle $S(t,t+1,v)$ designs; for details see [C32]. Moreover, the recursive doubling behaviour of the quasigroup isomorphism procedure carries over naturally to handle isomorphism problems for many classes of 1-factorizations in subexponential time. Consequently, there exist subexponential isomorphism algorithms for 1-factorizations of arbitrary connected graphs, 1-factorizations of complete multigraphs, Room squares and Howell designs [C11]. There also exist subexponential time isomorphism algorithms for Hadamard matrices [C12] and symmetric designs [L18]. In the case of symmetric designs with $\lambda=1$, i.e. projective planes, Miller [M12] showed that isomorphism testing can be performed in $O(v^{\log\log v})$ time. Babai and Luks [B3] have since extended this result to show

Theorem 3.3 [B3]: Canonical forms (and hence isomorphism testing) for symmetric $B[k,\lambda;v]$ designs can be found in $v^{O(\log\log v)}$ time.

However, no infinite family of symmetric designs is known for any $\lambda>1$.

The more exciting result contained in Babai and Luks' paper [B3] concerns computing canonical forms for graphs of bounded valence in polynomial time. The canonical form problem for graphs is closely related to the problem of testing isomorphism; the second task can be performed at least as fast as the first and, in most instances, an isomorphism test for a class of graphs consists of a procedure for determining the canonical form. Hence, a fast algorithm for determining the canonical form of a class of graphs (or designs), implies a fast algorithm for isomorphism testing of that class. Babai and Luks [B3] establish

Theorem 3.4 [B3]: Canonical forms for graphs of maximum degree d can be computed in $O(n^{f(d)})$ steps where $n=|V(G)|$.

Theorem 3.5 [B3]: Canonical forms for $B[k,\lambda;v]$ designs can be computed in $v^{f(k,\lambda)+\log v}$ time.

In other words, isomorphism testing of $B[k,\lambda;v]$ designs with fixed k and λ can be done in subexponential time. Babai and Luks' results represent a major advance in the research concerning graph isomorphism. Moreover, from a design theory point of view, Theorem 3.5 is a nice contrast to the isomorphism completeness result for general block designs.

Another class of designs in which some improvement with regard to isomorphism testing might be expected is cyclic designs. There is an elementary polynomial time algorithm for deciding equivalence of two difference families. Hence, if all inequivalent designs are non-isomorphic, there would be a polynomial-time algorithm for deciding isomorphism of difference families. However, this is not the case.

Theorem 3.6 [B16]: There exist inequivalent, isomorphic B[3,2;v] designs.

The smallest known pair exists when $v=16$. Furthermore, Brand [B16] has established the existence of an infinite family of such designs. For even values of n, Brand [B16] constructs $2^{(n/2)-1}$ cyclic designs on Z_{2^n} which are not equivalent as cyclic designs. He further establishes that these designs can be paired off so that the designs in a pair are isomorphic. However, no pair of inequivalent isomorphic STS is known, despite the fact that there exist cyclic STS which have non-multiplier automorphisms. The Bays-Lambossy theorem [B6, L1] guarantees that such a pair does not exist on a prime order; for details of the theorem, see [B6 PartII; C36].

Theorem 3.7 [B6 PartII]: Given 2 isomorphic cyclic structures on a prime number of elements, there exists a multiplier isomorphism transforming one to the other.

Theorem 3.7 is a statement about cyclic hypergraphs, a broad class of structures incorporating both circulants and cyclic designs. Using this theorem, we observe that there is an $O(v^2)$ algorithm for deciding isomorphism of cyclic designs with a prime number of elements. In deciding this complexity, we assume that the algorithm is given a cyclic representation of each design; the complexity of recognizing cyclic designs is unknown to the author. In practice, this does not create any difficulty since one usually deals with a difference family representation of the design.

There remain several interesting open questions regarding isomorphism testing of block designs. In the case of cyclic designs, the main question is whether there exists a pair of inequivalent, isomorphic STS. Ideally, one would like to prove that such a pair does not exist. Or perhaps, the Bays-Lambossy theorem can be extended to the case of STS(v) where v is the product of two primes. As it has now been established that isomorphism testing is subexponential for several classes of block designs, it would be interesting to see if any of these results can be extended to include other classes of designs or to establish a polynomial time algorithm for any non-trivial class of designs.

3.2 Isomorphism Invariants

The lack of a polynomial time algorithm for block design isomorphism compels us to search for other techniques which reduce the magnitude of this problem. In particular, given a list of designs, we require a method of partitioning the list into classes such that two isomorphic designs are in the same class. A design property which partitions the list in such a way is an isomorphism invariant. We view an ***invariant*** as a function I for which $I(D_1)=I(D_2)$ if D_1 and D_2 are isomorphic. When $I(D_1)=I(D_2)$ if and only if $D_1 \neq D_2$, the invariant is ***complete***. There is no known efficiently computable complete invariant, for designs in general. To maintain efficiency in resolving isomorphism we must, at present, resort to incomplete invariants. In choosing an invariant we wish to reduce the magnitude of the problem as much as possible. With this in mind, Petrenyuk and Petrenyuk [P3] propose that a measure of the invariant's effectiveness be its ***sensitivity*** -- the ratio of the number of classes it distinguishes to the number of non-isomorphic designs under consideration. A complete invariant has sensitivity one. In the remainder of this section, we consider invariants with respect to ease of computation and sensitivity.

3.2.1 Invariants for Block Designs

One of the earliest invariants employed was the order of the automorphism group. This invariant, however, is insensitive. A second difficulty is that no polynomial time algorithm is known for computing the order of the automorphism group. In fact, there is evidence that computing the order of the automorphism group is equivalent to deciding isomorphism; in the related case of graphs, the problem is isomorphism complete [B1, M3].

Another means of distinguishing designs is by examining the number and type of subdesigns. Moore [M13] used this invariant to demonstrate the existence of at least two non-isomorphic STS, $v > 13$. This invariant is also insensitive. Again, there is no known polynomial time algorithm for deciding whether one design is a subdesign of another. The corresponding problem for graphs is NP-complete.

Of course, there is no reason why one cannot employ subcomponents other than subdesigns as invariants. For example, Gibbons [G5] used fragments to distinguish various STS; this approach is discussed later in this section. Another possibility is to employ information concerning parallel classes or partial parallel classes. For example, one might consider the number of distinct parallel or partial parallel classes or various intersection patterns of such classes; these approaches are discussed in section 5 of this paper.

One invariant for general block designs, which has been successfully employed by several researchers, is clique analysis. Given a design D, we can define a series of ***block intersection graphs*** G_i, $i=0,...,k$, defined as follows:

The vertices of G_i are the blocks of D. Two vertices are adjacent if and only if the corresponding blocks contain exactly i elements in common.

One effective invariant is the number of cliques of size c in G_i; this is referred to as (c,i)-clique analysis. Gibbons [G5] employed clique analysis to help distinguish B[3,1;15], B[3,2;9], B[4,3;8], B[4,3;9] and B[5,3;10] designs. In distinguishing B[3,2;9] designs, Mathon and Rosa [M6] also used clique analysis, as cycle structure (which we discuss shortly) does not suffice. For cyclic STS, $v \leq 27$, (4,0)-clique analysis is a complete invariant [C36]. The complexity of this invariant is also appealing; an $O(b^4)$ algorithm for computing this invariant is immediate. When the design is transitive, we need only consider the number of cliques containing a particular element. Hence, an $O(rb^3)$ algorithm results. However, the number of cliques for relatively small values of v is enormous; for example, one of the S(2,3,21) designs contains 24646 (4,0)-cliques [C36]. Hence, although the growth is polynomial, the computation is extremely expensive. Furthermore, it appears that in order to maintain high sensitivity, the size of cliques being examined must increase as a function of v. If this is indeed the case, the computation is extremely difficult from a complexity standpoint -- it is, in fact, a special case of a #P-complete problem [G4, V1].

Other design properties which can be used to distinguish non-isomorphic designs include both the chromatic number and the chromatic index; these are discussed further in section 4.

3.2.2 Invariants for S(t,t+1,v)

In 1913, White [W1] introduced a method of distinguishing the two S(2,3,13) designs. Given a STS D, consider a triad (x,y,z) which is not in D. (x,y,z) is transformed by replacing each pair (x,y), (x,z), (y,z) by the single element with which it appears in D. Another triad results. For example, let D contain the three triples (1,2,4), (1,3,5), (2,3,6); the triad (1,2,3) will be transformed into (4,5,6). If one continuously repeats this operation, one of two things must occur. Either a triad of D is encountered or a previous triad is again reached. For simplicity, White refers to triads of D as one term cycles. Hence, every triad not in D initiates a *train* of triples which terminates in a periodic cycle. Trains are a special class of transformation graphs; for a more general study of transformation graphs, see [D3, D4]. Examining these trains, White differentiated the two S(2,3,13) designs. Although White proposed this invariant simply for STS, the obvious extension allows one to construct trains for $S(t,t+1,v)$ designs in $O(v^{t+1})$ time.

The train of a $S(t,t+1,v)$ design is a directed graph in which each component is a special *tree-like* directed graph. With this in mind, we can employ the optimal linear time tree isomorphism algorithm [C10, H13] in conjunction with Booth's optimal labelled cycle isomorphism algorithm [B12] to obtain an optimal algorithm for deciding isomorphism of trains. These observations supply us with a practical and efficient isomorphism method for trains [C22] which we would like to use to distinguish $S(t,t+1,v)$ designs.

The question is: how sensitive are trains? Trains successfully distinguish all eighty STS(15). In fact, their structure varies dramatically, and hence there is every reason to expect that they are a useful invariant for larger STS. One piece of theoretical evidence which supports this is the fact that every outregular directed graph with outdegree 1 not containing a cycle of length two appears as a subgraph of the train of some STS. However, trains do not completely distinguish nonisomorphic STS:

Lemma 3.8 [C22]: There are nonisomorphic STS having isomorphic trains.

Proof: Consider the Hall triple systems, in which every three elements generate either a block or a sub-STS(9). The train of such an STS consists simply of copies of the train of the STS(9), and depends only on the order v. But there are non-isomorphic Hall triple systems of order 3^m for all $m \geq 4$ [H1].

One serious problem with trains is their size; the graph contains $\binom{v}{3}$ nodes. A smaller invariant is desired. Retaining just the number of components is not enough, nor is retaining the component sizes, since the trains of the first seven STS(15) from [G5] all consist of 35 components of 13 vertices each. With the additional information of the number of sources (vertices of indegree zero) in each component, all STS(15) are distinguished except designs 6 and 7 [C22]. Although this simplified invariant, a *compact train*, is easy to compute and requires little storage, it is unclear whether they retain sufficient power.

Stinson [S4] instead examines the indegree sequence of trains.

Lemma 3.9 [S4]: No vertex in a train has indegree exceeding $v-2$. Further, any vertex of indegree $v-2$ is a block of the STS.

Since the indegrees are at most $v-2$, we may form a vector $(a_i : 0 \leq i \leq v-2)$, where a_i is the number of vertices of indegree i. We refer to this as the *indegree list* of the train. The space required to store an indegree list is clearly proportional to v, so we have a "small" invariant. Of course, the time required to compute the invariant is still proportional to v^3. For STS(15), indegree lists distinguish all non-isomorphic designs; the lists are presented in [S4].

Another invariant introduced to distinguish STS is cycle structure, which is sometimes referred to as the graph of interlacing. Several researchers have employed cycle structure to distinguish triple systems of small orders [C32, C37, C42, H3, M6, M15, P3, S4]. We describe it here in a more general setting [C36].

For a given $S(t,t+1,v)$ system $D=(V,B)$, consider any set $A \subset V$ such that $|A| = t-1$. For convenience, let $A=\{x_1,x_2,...,x_{t-1}\}$. We define a graph G_A to be $G(V_A,E_A)$ where $V_A=V-A$ and

$$E_A=\{(a,b) \mid a,b \in V_A, <x_1,x_2,...,x_{t-1},a,b> \in B\}.$$

This graph is a 1-factor.

Given D, consider two sets of elements $A=\{x_1,x_2,...,x_{t-1}\}$ and $C=\{x_1,...,x_{t-2},x_t\}$. We define $G_A=(V_A,E_A)$ and $G_C=(V_C,E_C)$ as above. We now define the union of two such graphs $G_A \cup G_C$ to be $G(V',E',L)$ where

$$V'=V_A \cap V_C - \{x \mid <x_1,...,x_t,x> \in B\}$$

and

$$E'=\{(a,b) \mid a,b \in V',(a,b) \in E_A or (a,b) \in E_C\}$$

and L is a mapping of edges to labels. $L(a,b)=A$ if $(a,b) \in E_A$. Because every t-tuple must appear exactly once in D, each element x in V' appears once in a block with the set A and once with the set C. Hence, $G_A \cup G_C$ is regular of degree 2; it is therefore a union of cycles.

A compact notation for this graph is just the list of cycle lengths in ascending order. This is called the *cycle list* for the pair of $(t-1)$-sets A and C. Consider the cycle lists for every pair of $(t-1)$-sets, which have $t-2$ elements in common. This collection of lists, when ordered lexicographically, is called the *cycle structure*. For cyclic STS, one only has to consider the cycle lists for the pairs $(0,i)$, $1 \leq i \leq (v-1)/2$.

In order to estimate the sensitivity of this invariant, the author [C32, C36] employed it to distinguish cyclic STS(v), $v \leq 45$; for these designs, cycle structure's sensitivity is approximately 0.9. For SQS, this invariant has been used by Phelps [P7] to distinguish the twenty-nine S(3,4,20) designs.

There is an elementary $O(v^3)$ algorithm for computing this invariant for STS ($O(v^2)$ for cyclic STS). Its high sensitivity guarantees the existence of many classes containing a single design. It has the added attraction that even for classes containing more than one design, a subexponential isomorphism algorithm based on cycle structure can be employed to differentiate the designs [C32].

Like trains, one difficulty with cycle structure is the space requirement. Gibbons [G5] suggested a way of compressing the cycle structure by considering only cycles of length 4. Instead of keeping the list of cycle lengths for $G_A \cup G_C$, simply count the number of cycles of length 4. By keeping this information for each pair of elements, one forms the *fragment vector* for the STS.

Note, we do not have to determine all the graphs $G_A \cup G_C$ in order to find the fragment vectors. A *fragment* is a set of four blocks of the form (u,v,w), (u,x,y), (v,x,z), (w,y,z). A fragment gives rise to a 4-cycle in $G_u \cup G_z$, $G_v \cup G_y$

and $G_w \cup G_x$. We can determine the fragment vector simply by finding all fragments, and each time one is encountered, updating the fragment vector appropriately. Although this method still requires time proportional to v^3, it is considerably quicker than determining the complete cycle structure. It also has the added advantage of requiring less space than cycle structure; the fragment vector requires space proportional to v.

Gibbons [G5] used fragment vectors to distinguish all 80 STS(15). In [S4], Stinson compared the sensitivity and efficiency of indegree lists and fragment vectors on random STS(v), $15 \le v \le 31$, generated via a hill-climbing algorithm [S5]. Both invariants are complete for STS(v), $v \le 15$; for larger v, Stinson concludes that both invariants seem to be very successful in practice. Both invariants can be computed in time $O(v^3)$ and require space $O(v)$. Experimental evidence suggests that the invariant based on trains is more effective, but it requires about five times longer to compute [S4].

For further information regarding many of the aforementioned invariants and other properties for specific STS, see [M5].

Stinson and Vanstone [S8] in their examination of nonisomorphic Kirkman triple systems, developed an invariant which exploits information concerning the design's resolution. Consider a KTS($6t+3$) (V,B) with a resolution $R=\{R_1,...,R_{3t+1}\}$. If (x,y,z) is a block, define $other(x,y)=z$ and $rc(x,y)=R_i$ if $(x,y,z) \in R_i$. Now define a partial mapping g from the 3-subsets of V to the 3-subsets of R. If x, y and z are distinct members of V, let $z_1=other(x,y)$, $y_1=other(x,z)$ and $x_1=other(y,z)$. If (x_1,y_1,z_1) is not a block, define $g((x,y,z))=\{rc(x_1,y_1),rc(x_1,z_1), rc(y_1,z_1)\}$. For $i \ge 0$, let f_i denote the number of 3-subsets of R which have an inverse image of cardinality exactly i. Finally, define $INV(R)=(f_i \mid 0 \le i \le v)$. $INV(R)$ is an invariant for Kirkman triple systems. Stinson and Vanstone employ this invariant to distinguish nonisomorphic KTS(39) and KTS(51) [S8].

The construction of the above KTS is based on strong starters; as noted earlier, strong starters have been successfully used to construct a variety of combinatorial configurations including Room squares and Howell designs. For appropriate algorithmic techniques for generating inequivalent or nonisomorphic strong starters in cyclic groups, see [K5].

3.2.3 Invariants for Steiner Systems

In the previous section, we defined cycle structure, which is applicable only when $k-t=1$. However, when this is not the case, we can still define the graph G_A, $|A|=t-1$. G_A is a collection of disjoint $(k-t+1)$-cliques. We may again define the labelled graph $G_A \cup G_C$, as before. Any invariant of this graph is an invariant of the pair of $(t-1)$-sets A and C. For a given invariant I, let $I(A,C)$

denote the value of I on $G_A \cup G_C$. An invariant of the design is the multiset

$$\{I(A,C) \mid |A|=t-1, |C|=t-1, |A \cap C|=t-2, A \subset V, C \subset V\}.$$

One can see that cycle structure is an invariant of this form. Let us consider a specific graph $G_A \cup G_C$. Let X be the $(k-t+1)$-clique common to both G_A and G_C. The $(k-t+1)$-cliques of $G_A - X$ can be arbitrarily ordered. Then the $(k-t+1)$-cliques of $G_C - X$ can be represented in terms of the cliques of $G_A - X$ e.g. a $(k-t+1)$-set $S(K)$; if v belongs to the ith clique of $G_A - X$ and $v \in K$, $i \in S(K)$. Observe that for $v,w \in K, v \neq w$, v and w belong to different cliques in $G_A - X$. Hence, $S(K)$ is a $(k-t+1)$-set. For a given i, consider the $k-t+1$ sets $S(K_1),...,S(K_{k-t+1})$ which contain i. From this collection form $T(K_j)=S(K_j)-\{i\}$. Now $T(K_1),....T(K_{k-t+1})$ form the edges of a $(k-t)$-uniform hypergraph, which we will denote H_i and call an *overlap graph*.

Any invariant of the collection $\{H_i\}$ is an invariant of $G_A \cup G_C$. Each overlap graph H_i has the same number of edges, so this invariant would result in no discrimination. However, they may have a different number of vertices. With this in mind, we define the overlap list of $G_A \cup G_C$, $OL(A,C)$, to be the multiset $\{|V(H_i)|\}$. The overlap list is clearly invariant under isomorphism. The *overlap structure* of a design is the multiset

$$\{OL(A,C) \mid |A|=t-1, |C|=t-1, |A \cap C|=t-2, A \subset V, C \subset V\}.$$

A seemingly more powerful invariant can be defined by enumerating all $(k-t)$-uniform hypergraphs with $(k-t+1)$ edges and arbitrarily ordering them 1 through m. For such a hypergraph H, denote by $I(H)$ its index in this list. The *typed overlap list* of $G_A \cup G_C$, $TOL(A,C)$, is the multiset $\{I(H_i)\}$. The *typed overlap structure* is the obvious analogue of overlap structure.

With respect to computation, there is an efficient algorithm for computing this invariant [C33]. Furthermore, the invariant appears to be quite sensitive. For example, overlap structure distinguished all cyclic S(2,4,v) designs, $v \leq 64$, and all cyclic S(2,5,v) designs, $v \leq 65$ [C33, C36].

4. Colouring Block Designs

4.1 Colouring Elements

An *r−colouring* of a hypergraph is an assignment to each vertex of a colour chosen from an *r−set* of available colours; equivalently, it is a partition of the vertices into r sets. An r-colouring is *proper* if no edge contains solely vertices of one colour. A hypergraph is *r−colourable* if it has a proper r-colouring, and is *r−chromatic* if it r-colourable but is not $(r-1)$-colourable. The *chromatic number* of a hypergraph H, denoted $\chi(H)$, is that r for which H is r-chromatic.

Many researchers have examined colouring graphs and hypergraphs; in fact, these problems arise in many areas of computer science [C2, E3]. We focus here upon the analogous colouring problems for combinatorial designs. For another survey of results concerning the colouring of Steiner systems, the reader can also refer to [B10]. There are many reasons for examining such colouring problems and related tasks. Investigations of the chromatic number have led both directly and indirectly to elegant constructions for t-designs; one recent example is the investigation of 2-chromatic SQS [P11]. Furthermore, colouring information is a means of distinguishing designs and is, of course, an isomorphism invariant. Unfortunately, determining the chromatic number of a design appears to be a computationally difficult task, and hence this is not a practical invariant.

This does however raise another motivation for examining colouring problems. It is well-known that deciding whether a graph is k-colourable (for fixed $k \geq 3$) is an NP-complete problem [G4]. Determining whether a graph is 2-chromatic can be easily carried out in linear time -- we need only decide if the graph is bipartite (see [C2], for example). On the other hand, deciding whether a hypergraph is 2-chromatic is NP-complete [L17]. Do such problems remain NP-complete when one is examining block designs or Steiner systems? Or can the structure of block designs be exploited to ensure polynomial time algorithms? We will examine some of these questions in this section.

First, we present Lovász' NP-completeness result regarding colouring hypergraphs. The construction is presented here as an example of the type of transformation which is required in such proofs.

Theorem 4.1 [L17]: Deciding whether a 3-uniform hypergraph is 2-colourable is NP-complete.

Proof:

Membership in NP is immediate. To show completeness, we give a polynomial time reduction from the problem of graph 3-colourability. Given a graph $G=(V,E)$, $V=\{v_1,\ldots,v_n\}$, we define a 3-uniform hypergraph $H=(W,F)$. The vertex set, W, is $\{\infty\} \cup \{x_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq 3\}$. The edges in F are

(1) $\{\infty, x_{ik}, x_{jk}\}$ for all $\{v_i, v_j\} \in E$, $1 \leq k \leq 3$.

(2) $\{x_{i1}, x_{i2}, x_{i3}\}$ for $1 \leq i \leq n$.

Now H is 2-colourable if and only if G is 3-colourable.

A 3-uniform hypergraph can be transformed into a partial SQS H such that H is 2-colourable if and only if the original hypergraph was 2-colourable. Furthermore, the transformation can be performed in polynomial time which establishes

Theorem 4.2 [C20]: Deciding whether a partial SQS is 2-colourable is NP-complete.

As an interesting contrast to the previous result, it has been established that

Theorem 4.3 [C20]: Deciding whether a SQS is 2-colourable can be performed in polynomial time.

To illustrate this, we follow the presentation in [C20]. A 2-colouring of a SQS (V,E) is a partition of V into two sets V_1 and V_2. It is proper if for any $e \in E$, $e \cap V_i \neq e$. Doyen and Vandensavel [D9] proved that if $<V_1,V_2>$ is a proper 2-colouring of (V,E), then $|V_1| = |V_2|$.

The algorithm operates by extending a partial colouring, which is a partition of V into three sets V_1, V_2 and U. Vertices in V_1 (V_2) have been assigned the first (second) colour; the colours of vertices in U are as yet unspecified. A partial colouring $<V_1,V_2,U>$ is *feasible* if there is a proper 2-colouring $<W_1,W_2>$ for which $V_1 \subseteq W_1$ and $V_2 \subseteq W_2$. All feasible partial colourings are proper, but of course the converse need not hold.

A simple-minded method which uses Doyen and Vandensavel's observation is the following. Given a partial colouring $<V_1,V_2,U>$ first check that it is proper. If it is not, it is not feasible. Next check if either $|V_1|$ or $|V_2|$ is $|V|/2$; if so, we have completed a proper 2-colouring [D9]. In the final case, we attempt to extend the partial 2-colouring. For each $v \in U$ in turn, we determine whether $<V_1 \cup \{v\},V_2,U-\{v\}>$ is feasible. If any one of these is feasible, $<V_1,V_2,U>$ is feasible; otherwise it is not.

Now a SQS (V,E) is 2-colourable if and only if $< \emptyset , \emptyset , V>$ is feasible. When extending a partial colouring, additional information can be exploited. For example, if $\{w,x,y,z\}$ is an edge (block) for which $w,x,y \in V_1$ and $z \in U$, we know that z must be placed in V_2. Therefore, we say that z is an *implicant* for V_2 (V_1) if $z \in U$ and there is an edge $\{w,x,y,z\}$ with $w,x,y \in V_1$ (V_2).

To circumvent the selection of vertices leading immediately to improper colourings, we introduce a process called *stabilization*. Given a partial colouring $<V_1,V_2,U>$, we locate the set $U_1 \subseteq U$ of implicants for V_1 and the set $U_2 \subseteq U$ of implicants for V_2. If U_1 and U_2 contain an element in common, a proper colouring is impossible, in which case the stabilization has *failed*. Otherwise, if $U_1=U_2=\emptyset$, stabilization is said to *succeed*. If the stabilization process has neither failed nor succeeded, we repeat the process and stabilize $<V_1 \cup U_1, V_2 \cup U_2, U-U_1-U_2>$.

Stabilization can be carried out in polynomial time, and thus it can be used to substantially improve the simple-minded algorithm mentioned earlier. After each selection, we stabilize the partial colouring and then attempt to extend the resulting partial colouring. In fact, we need only deal with stable partial colourings throughout the algorithm.

To guarantee an improvement over the exponential running-time, we need two additional facts. The first concerns the sizes of the two colour classes in a stable colouring. If $<V_1, V_2, U>$ is a stable partial colouring of a SQS (V,E), $|V_1|-2 \leq |V_2| \leq |V_1|+2$. Secondly, one needs to establish that at each step of the algorithm, one can select an element to colour which has a sufficiently large number of implicants. Hence, the algorithm (or this step of the algorithm) cannot be invoked too often. In fact, it can be invoked at most O(log n) times; hence, the algorithm runs in polynomial time. Once this fact is established, one can apply a greedy selection process, which results in a polynomial-time algorithm.

Although, this result has been presented here for SQS, it can clearly be generalized to other families of t-designs in which one can exploit the existence of implicants. In other words, deciding whether a t-B$[t+1,\lambda;v]$ design ($t \geq 3$) can be 2-coloured can be performed in polynomial time. Doyen and Vandensavel's result indicates that only designs with an even number of elements can be 2-coloured. Therefore, no STS can be 2-coloured.

Projective planes, with the exception of the STS(7), can be 2-coloured. Given a 2-colouring, the smaller colour class is called a *blocking set*. For results concerning blocking sets in designs, see [B15, B18, D5, D11].

The existence of k-chromatic STS for $k \geq 3$ has been examined [D1, R6, R7]. In particular, Rosa [R7] established the existence of a 3-chromatic STS of all admissible orders. A more recent paper [D1] established a much more general result:

Theorem 4.4 [D1]: For any $k \geq 3$, there exists an n_k such that for all admissible $v \geq n_k$ there exists a k-chromatic STS of order v.

Furthermore, de Brandes, Phelps and Rödl [D1] established that $n_4 \leq 49$. In so doing, two colour-preserving recursive constructions are presented.

Theorem 4.5 [D1]: If there exists a k-chromatic STS(v), there exists a k-chromatic STS($2v+1$).

Theorem 4.6 [D1]: Let $v \equiv 1,9 \pmod{12}$. If there exists a k-chromatic STS(v), there exists a k-chromatic STS($2v+7$).

In the course of their examination of k-chromatic STS, de Brandes, Phelps and Rödl raise some very interesting existence algorithmic questions. For example, do there exist uniquely colourable k-chromatic STS for all k? A corresponding question which one might ask is "Given a k-colouring of a k-chromatic STS, how difficult is it to establish that this colouring is unique?".

Before examining this question, it is sensible to answer the more basic question "How difficult is it to decide whether a STS is k-chromatic?".

It is known that

Theorem 4.7 [P10]: Deciding whether a STS is k-chromatic is NP-complete.

There are several related results which warrant mention here, the first of which concerns partial STS.

Theorem 4.8 [C21]: Deciding whether a partial STS is t-colourable is NP-complete for any fixed $t \geq 3$.

Proof:

In order to prove this theorem, we construct t-chromatic partial STS in which any t-colouring assigns a fixed pair of elements different colours. To do this, we need the following two lemmas.

Lemma 4.9: For each $t \geq 2$, there is a t-chromatic partial STS for which any t-colouring assigns the same colour to two fixed elements.

Proof:

There are $(t+1)$-chromatic STS for all $t \geq 2$ [D1]. Suppose P is a $(t+1)$-chromatic STS. A triple is said to be *critical* if its deletion lessens the chromatic number of the partial STS. Starting with any $(t+1)$-chromatic system, we delete blocks until one becomes critical. Call this partial STS P. Deleting a critical block from P produces a t-chromatic partial STS P'. Any t-colouring of P' assigns the same colour to the three elements forming the critical block of P, since otherwise the t-colouring of P' would also t-colour P, which is in contradiction to our assumption.

Lemma 4.10: For each $t \geq 2$, there exists a t-chromatic partial STS P and a fixed pair of elements $\{x, x'\}$ of P, such that any t-colouring of P assigns a different colour to x and x'.

Proof:

Let P be a partial STS with chromatic number t, having the property that any t-colouring of P assigns the same colour to two given elements x and y. Denote the element set of P by $Q \cup \{x\}$. Take two copies of P, one on $Q_1 \cup \{x\}$ and one on $Q_2 \cup \{x\}$ that is, two copies intersecting only at x. Add a new element x' and include the block $\{y_1, y_2, x'\}$. This partial STS is t-chromatic and any t-colouring must assign the same colour to x, y_1, and y_2. Then x' must be coloured differently from x.

Proof of Theorem 4.8:

Suppose we are to decide whether an arbitrary graph G is t-colourable; we know that this problem is NP-complete for any fixed $t \geq 3$ [G4]. First, let P be a partial STS with chromatic number t having fixed elements x, x' which every t-colouring P assigns two different colours. We construct a partial STS with a copy of P for every edge of the graph G; for an edge $\{y, z\}$ of G, we take a copy of P disjoint from the other copies, and identify x and x' with y and z. The theorem follows directly.

Given a partial STS on v elements, one can produce in polynomial time a $B[3,12tv+3;18tv+3]$ design which is $3t$-colourable if and only if the partial STS is t-colourable. Hence, there is a polynomial time reduction of a known NP-complete problem to a more general colouring problem, establishing that

Theorem 4.11 [C21]: Deciding whether a block design is t-colourable is NP-complete for all $t \geq 9$.

However, the above result is established for block designs with relatively large λ; in fact, λ is $O(v)$. However, even restricting one's attention to small fixed λ does not necessarily result in any improvement. Phelps and Rödl [P10] more recently established that

Theorem 4.12 [P10]: Deciding whether a STS is 14-colourable is NP-complete.

To establish this result, Phelps and Rödl employed the fact that deciding 3-colourability of 4-regular graphs is NP-complete [G4]. Given a 4-regular graph G, a partial STS P is constructed such that the chromatic number of P is four times that of G. The partial STS P is then embedded into a STS S. Moreover, the chromatic number of S is at most $\chi(P)+2$. Therefore, if $\chi(G) \leq 3$, then $\chi(P) \leq 12$ and $\chi(S) \leq 14$. Alternatively, if $\chi(G) \geq 4$, $\chi(P) \geq 16$. In order to establish the NP-completeness results, it is necessary to guarantee that the embedding can be performed in polynomial time, which is indeed the case. The embedding employed transforms a partial STS(v) into a STS($6v+3$) and is done in polynomial time.

Related work concerns the existence of particular colourings. For example, given a block design in which the elements are coloured with m colours and the colouring is proper, can one produce a complete colouring with $m+1$ colours? A colouring is *complete* if the merging of any two colour classes would result in an improper colouring e.g. a monochromatic block. Cockayne, Miller and Prins [C4] have proved several interesting results along these lines. First let us define what is meant by a type 1 colouring. A colouring $\{V_1, V_2, \cdots V_n\}$ is said to be *type* 1 if, for all $x \in V_i$ and all $j<i$, $\{V_1, \ldots V_j \cup \{x\}, \ldots, V_i - \{x\}, \ldots V_n\}$ is an improper colouring. In other words, no element can be moved to a "lower" colour class.

Theorem 4.13 [C4]: Given a design $D=(V,B)$ with complete type 1 colourings of orders m and n $(m<n)$, one can obtain a complete type 1 colouring for each order p, $m<p<n$.

Proof:

Let $Q=\{Q_1, \cdots Q_m\}$ and $R=\{R_1,...,R_n\}$ be the complete type 1 colourings of order m and n, respectively. Now consider the partitioning $Q^1=\{R_1,Q_1-R_1,Q_2-R_1, \cdots Q_m-R_1\}$. This is a type 1 complete colouring of V, using at most $m+1$ colours. Similarly, $Q^2=\{R_1,R_2,Q_1-R_1-R_2, \cdots Q_m-R_1-R_2\}$. Repeating this process, we obtain a sequence of type 1 complete colourings of V: $Q,Q^1,Q^2,...,Q^n=R$. The number of colours employed in Q^{i+1} is either the number of colours employed in Q^i or one more.

In fact, Cockayne, Miller and Prins [C4] prove a more general result regarding the existence of colourings; first we require some further definitions.

Instead of asking what is the least number of colours required to colour the elements of a design, one could examine the achromatic number of a design. Given a t-B[$k,\lambda;v$] design D consider a n-colouring i.e. a partition of the elements into disjoint sets V_1, V_2, ... V_n such that each V_i is an independent set of D. That is to say that for all i, no k elements of V_i constitute a block. Furthermore, for each pair of distinct sets V_i, V_j, $V_i \cup V_j$ is not an independent set. The *achromatic number* of D, denoted $\psi(D)$, is the largest n such that D can be n-coloured according to the above conditions. Deciding whether a graph is k-achromatic is NP-complete [G4]. The complexity of deciding whether a design is k-achromatic is not known.

Theorem 4.14 [C4]: Given a design D, there exists a complete n-colouring for all n, $\chi(D) \le n \le \psi(D)$.

Sketch of Proof:

The proof is based on three observations and is again algorithmic in nature:

(1) Theorem 4.13 above.

(2) Given the "smallest" colouring e.g. one with $\chi(D)$ colours, one can obtain a colouring of the same size, which in addition to being complete, is also a type 1 colouring.

(3) Given a n-colouring which is complete (but not type 1), one can produce a type 1 (and hence complete) n-colouring or a complete $n-1$-colouring.

Nešetřil, Phelps and Rödl [N1] examined the achromatic number of STS and partial STS. In fact, the paper includes an algorithm for colouring a STS(v) with at least $c\sqrt{v}$ colours, obtaining a complete colouring. In order to establish this result, Nešetřil, Phelps and Rödl establish that any partial STS(n) with at least cn^2 blocks contains an induced subgraph which is quite dense (i.e. a set of blocks such that every element is contained in at least $cn/10$ blocks. The algorithm provided identifies this dense subgraph in polynomial time. Next, they present an algorithm which will employ at least $c\sqrt{n}$ colours to colour the subgraph. The algorithms will require at most a polynomial number of iterations; however, the maximum time required for an arbitrary iteration is unclear. It would be interesting to determine the time bound for this algorithm and/or establish a polynomial algorithm to produce colourings which require a large number of colours. No such algorithmic results have been established for other families of block designs.

The problems of colouring designs, locating independent sets, and determining subdesigns are related. One can think of any proper n-colouring as a partitioning of V into n disjoint independent sets, and independent sets can be viewed as the opposite of subdesigns. Independent sets in STS have been studied by de Brandes and Rödl [D2]. Despite the correspondence, the complexity of deciding whether a block design has an independent set of size at least k is unknown. The corresponding problem for graphs is NP-complete [G4].

The polynomial time algorithm for recognizing 2-chromatic SQS presented in [C20] is also an algorithm for deciding if a SQS of order $2n$ has a maximum independent set of size n, which is the largest possible [D9].

Another related problem is establishing the size of a design's smallest dominating set. Given a design $D=(V,B)$, a subset $V' \subseteq V$ is a *dominating set* if for all $b \in B$, there exists $u \in b$ such that $u \in V'$. Again the corresponding problem for graphs is NP-complete [G4]. Domination in designs seems not to have been studied.

4.2 Colouring Blocks

So far, we have examined problems associated with the colouring of a block design's elements. Alternatively, one can assign colours to blocks. In a block-colouring, a colour class is a set of pairwise disjoint blocks. In the design of experiments where each block corresponds to a 'test', we can view disjoint blocks as tests which can be carried out simultaneously. A n-block colouring is a partition of the blocks into n colour classes; the *chromatic index* is the least n for which such a colouring exists. In our example, the chromatic index is precisely the time required for the entire experiment. Designs and related systems have been employed in the scheduling of tournaments [K4, M14, S1, M10 and references therein]. The same analogy holds; the chromatic index corresponds to the least number of rounds. Designs with small chromatic index have been studied under the guise of resolvable or nearly resolvable designs (for example, see [H9, R2 and references therein]). We address the topic of resolvability later.

In the case of simple graphs, Vizing's Theorem [V3] guarantees that the chromatic index is either δ or $\delta+1$, where δ is the maximum vertex degree. Arjomandi [A4] gives a clever polynomial-time method for constructing a $(\delta+1)$-colouring. Nonetheless, in 1981 Holyer [H12] proved that deciding whether the chromatic index of a graph is δ or $\delta+1$ is NP-complete. Similar results are lacking for designs.

The majority of research concerning the chromatic index of designs has focussed on Steiner 2-designs, particularly STS. One question of interest is: what is the upper bound on the chromatic index? For Steiner 2-designs, a relatively weak bound is obtained from Brooks' Theorem [B11] which guarantees that the chromatic number of the design's block intersection graph (and hence the chromatic index of the design) is at most $\frac{kv}{k-1}$. One reason to suspect that this bound is quite weak is that a conjecture of Erdös, Faber and Lovász [E2] would ensure an upper bound of v. In fact, in the case of cyclic Steiner 2-designs, it has been shown that the chromatic index is at most v [C13].

As far as algorithmic results are concerned, less is known for designs than in the corresponding graph case. The complexity of computing the chromatic index is unknown; the current best method involves backtracking which could require an exponential amount of time. Instead of employing a backtrack, a depth-first branch-and-bound algorithm could be implemented, in which one chooses the most promising partial colouring to extend (instead of simply trying to extend the most recent). Although such an algorithm is still exhaustive, in general its running time should prove faster than the traditional backtracking algorithm. C. Colbourn [C7] has implemented such an algorithm and tested its performance on the eighty STS(15).

To develop such an algorithm, one must first define what one means by 'promising' partial colouring. A partial colouring leading to an optimal colouring might be expected to have few colours and many blocks; at the very least, it should not have many colours and few blocks. With this in mind, we define the *priority* of a partial colouring to be $p-vc$, where p is the number of blocks in the partial colouring, and c is the number of colours.

The branch-and-bound algorithm starts with all blocks uncoloured; at any stage, it maintains a priority queue of partial colourings which are candidates for extension. A partial colouring of highest priority is removed from the queue.

Each of the partial colourings resulting from extending this partial colouring is added to the priority queue.

Two simple heuristics prove quite useful in improving this basic method. The first involves the elimination of solutions using too many colours, as follows. The priority of a colouring is largely determined by the number of blocks in it; this gives the algorithm a tendency to complete colouring early. Once we have completed a colouring, we can ignore all partial colourings using as many or more colours. This preserves the major advantage of a depth-first approach.

Once a partial colouring is selected to be extended, a second heuristic is employed to further limit the computational effort. To extend a partial colouring, select an uncoloured block, and try colouring it with each available colour in turn, including assigning it a new colour. For a given block, there will be certain colours which it cannot be assigned. Having selected a partial colouring, we are still free to select any uncoloured block to perform the extension. The second heuristic is to select a block which can be assigned the fewest number of colours. The goal here is to limit the number of partial colourings considered.

Although this algorithm could theoretically require an exponential amount of time (from what we know so far), it has performed well in practice [C7].

Because there is no known polynomial time algorithm for computing the chromatic index, one must instead investigate algorithms which are guaranteed to run in polynomial time but which may only give approximate answers. Two general classes of algorithms for approximating the chromatic index were studied in [C15]: greedy methods and hill-climbing methods.

One very simple approach is a block-by-block greedy algorithm. Initially, no blocks are coloured. The blocks are coloured one at a time. For convenience, let the different colours be represented by integers; when colouring a block the least possible integer is assigned such that the resulting colour class still contains disjoint blocks.

Alternatively, a greedy algorithm could proceed colour-by-colour. Having selected i colour classes, we select the $(i+1)st$ by taking a maximal colour class from among the remaining uncoloured blocks. By *maximal* we mean:

(1) every uncoloured block intersects the colour class

(2) there is no block whose deletion from the colour class enables the simultaneous addition of two uncoloured blocks.

Clearly, the first colour is assigned to the largest set of mutually disjoint blocks (i.e. a partial parallel class). Therefore, we know that this first colour class must contain at least $(v-1)/4$ blocks [L10].

Both greedy algorithms can be improved by introducing some simple heuristics and hence hill-climbing from the current colouring to a better or more-promising colouring. Obviously, the block-by-block greedy method is highly sensitive to the order in which the blocks are presented. Therefore, one heuristic is to check to see if ever the blocks of a colour class can be distributed. A second heuristic operates as follows. If there are two blocks b with colour i and d with colour j, which can be recoloured so that b has colour j and d has colour k, and if in so doing the vector of colour class sizes is lexicographically increased, we do so.

These two simple heuristics improve the performance of both greedy methods. If one selects the best colouring from the two resultant methods, the colouring produced for each of the eighty STS(15) is close to optimal. This can be seen in the table below, which gives the number of STS(15) coloured with 7,8,9 and 10 colours.

Chromatic Index	Size of Best Colouring Obtained			
	7	8	9	10
7	2	0	2	0
8		0	13	0
9			50	13

As in the case of graphs [H5], one could also consider the *achromatic index* of a design; the maximum number of colours that could be required in a block-colouring. Again, the complexity of determining the achromatic index is unknown. An obvious greedy algorithm for obtaining a block colouring which may require many colours is as follows. Initially, assign each block a different colour. Then join together any two disjoint colour classes, and hence eliminate a colour. Continue to do so until no pair of disjoint colour classes exists.

4.3 Nesting Block Designs

One problem which is a special case of block colouring is nesting. A *nesting* for a triple system $B[3,\lambda;v]$ (V,B) is an assignment to each block $b \in B$ of an element $e(b) \in V$, such that adjoining $e(b)$ to b for each block b produces a block design with block size 4. It is a simple matter to verify that nesting a $B[3,\lambda;v]$ produces a $B[4,2\lambda;v]$, and hence that nested triple systems can only exist when $\lambda(v-1) \equiv 0 \pmod 6$, $v \geq 4$. Several researchers have examined nested designs and related configuration; for example, see [C17, L12, L14, L15, L16, M9, M16, N2, S7].

First let us examine the case when $\lambda=1$; a nested STS can only exist for $v \equiv 1 \pmod 6$. Some of the work regarding nesting STS has actually concerned nesting cyclic STS. A cyclic STS has a *cyclic nesting* if each starter block can be assigned a fourth element to produce starter blocks for a B[4,2;v] design. The following condition on a set of starter blocks for a STS is equivalent to the existence of a cyclic nesting: the starter blocks can be chosen so that each element i or its complement $v-i$ appears in exactly one starter block.

This equivalence allows us to state a conjecture of Novák [N2]: every cyclic STS of order $6t+1$ has a cyclic nesting. Novák verified this conjecture for $t \le 5$. Little work has been done to address Novák's conjecture. Longyear [L14] notes, however, that Bose's construction [B14] for cyclic triple systems yields a cyclic nesting for a cyclic STS for every order v where v is a prime or prime power.

Another generalization of nested STS has been studied by Mendelsohn [M9]. A *perpendicular array of triple systems* of order v, denoted $PATS(v)$, is a $v(v-1)/6$ by 4 array. When any column is omitted, the result is a STS. Note that every column, therefore, forms a nesting for the STS which remains. Mendelsohn [M9] shows asymptotic closure for the existence of $PATS(v)$, thereby producing nested STS for every admissible order with finitely many exceptions. This closure is obtained by employing a direct product and PBD closure for $PATS$; both of these extend trivially to nested triple systems.

Other constructions for nested triple systems can be obtained by noting the relation between nested triple systems and perpendicular arrays. A *perpendicular array* of order n and depth s is a $n(n-1)/2$ by s array, denoted $PA(n,s)$; every two columns contain each unordered pair of an n-set exactly once. A nested triple system of order v produces a $PA(v,4)$, as follows. When block (i,j,k) has nesting element l, include the rows (i,j,k,l), (k,i,j,l) and (j,k,i,l) in the PA. The resulting $PA(v,4)$ is invariant under the column permutation (123)(4); this is the *conjugate invariant subgroup* of the PA. Mullin, Schellenberg, van Rees, and Vanstone [M16] give a singular indirect product for $PA(v,4)$; this product preserves the conjugate invariant subgroup, and hence, provides a singular indirect product for nested triple systems.

Longyear [L16] and Stinson [S7] take a different approach to the construction of nested designs by examining nested group divisible designs. A *group−divisible* design (GDD) is a triple (X,G,A), where X is a set of points, G is a partition of X into subsets (called *groups*), and A is a set of subsets of X (blocks), such that a group and a block contain at most one common point, and any two points from distinct groups occur in a unique block. A GDD with block size 3 is said to be nested if one can adjoin a fourth point to each block, so that every pair from distinct groups occurs in two blocks. Through his examination of GDD, Stinson has managed to prove that

Theorem 4.15 [S7]: There is a nested STS if and only if $v \equiv 1 \pmod 6$.

Proof:

The condition $v \equiv 1 \pmod 6$ is necessary. As nested designs of prime power orders are known to exist, nested STS(v) for $v = 7, 13, 19$, and 37 can be assumed. For any other $v \equiv 1 \pmod 6$, add one new point to a GDD, replacing each group by a nested STS(7) on the six points in the group and the new point.

To complete the proof, Stinson needed to determine the existence of the necessary GDD. This was accomplished via Wilson's Fundamental Construction for GDD [W2, W3, W4], establishing the following result:

Theorem 4.16 [S7]: If $v \equiv 0 \pmod 6$, $v \geq 24$, $v \neq 36$, there is a nested GDD with groups of size 6 and blocks of size 3.

In the case when $\lambda \equiv 0 \pmod 6$, self-orthogonal Latin squares are used to construct nestings; when $\lambda \equiv 3 \pmod 6$, Room squares are used and when $\lambda \equiv 2,4 \pmod 6$, almost resolvable twofold triple systems are used. Combining these various constructions together, it is possible to show:

Theorem 4.17: Nested triple systems $B[3,\lambda;v]$ exist whenever $\lambda(v-1) \equiv 0 \pmod 6$, $v \geq 4$.

5. Resolvability of Block Designs

Many researchers have examined the existence of resolvable designs; here, we examine some of the algorithmic issues which arise when discussing partial parallel classes (PPC) and resolvability. First, let us recast the problem in a different setting.

A PPC is simply a collection of mutually disjoint blocks of the design. Now instead, look at the block intersection graph; two vertices are adjacent if their corresponding blocks share at least one element in common. Therefore, a PPC in the design corresponds to an independent set in the block intersection graph; hence, determining whether a design is resolvable involves finding a partition into maximum independent sets. However, it is well-known that finding the maximum independent set of a graph is NP-complete [G4]. If one views the task in terms of hypergraphs, a PPC has also been termed a matching.

Although the complexity of determining whether or not a design is resolvable is unknown, the problem is likely to be NP-complete. Therefore, it is unlikely that there exists an efficient algorithm for determining the resolutions of a design. The usual approach is to employ a backtracking algorithm to determine all of the design's parallel classes and then attempt to piece these classes together (again via backtracking) to form a resolution.

Perhaps we can hope for some improvement in a restricted class of designs. For example, consider a Steiner 2-design; the block intersection graph of a Steiner 2-design is a strongly regular graph, i.e. each pair of adjacent vertices has the same number of common neighbours and each pair of non-adjacent vertices has the same number of common neighbours. The converse is also true; for sufficiently large v, strongly regular graphs with the appropriate parameter sets, are block intersection graphs of some $S(2,k,v)$ [P5]. Unfortunately, the complexity of finding a maximum independent set in a strongly regular graph is unknown.

The majority of research concerning the resolvability of designs has focussed upon triple systems. It is well-known that resolvable or Kirkman triple systems (KTS) exist for all orders $v \equiv 3 \pmod 6$ [R2]. In the case $v \equiv 1 \pmod 6$, a parallel class is, of course, impossible. However, one can hope to partition the system to form a nearly Kirkman triple system. In either case, one is forming a collection of PC or PPC of size $v/3$ when $v \equiv 3 \pmod 6$ or $(v-1)/3$ for $v \equiv 1 \pmod 6$. Of course, it is not always possible to form a PPC which is this large. Hence, an obvious question is "Given a STS(v), what is the size of the largest PPC?"

Several researchers have established lower bounds in an attempt to answer this question [B17, L10, P4, P5, W5]. Let $t-\pi[k,\lambda;v]$ be the largest number such that every t-B$[k,\lambda;v]$ design has a PPC containing $t-\pi[k,\lambda;v]$ blocks. Lindner and Phelps [L10] proved that

Theorem 5.1 [L10]: $t-\pi[t+1,1;v] \geq (v-t+1)/(t+2)$, where $v \geq t^4+3t^3+t^2+1$.

Woolbright [W5] has since improved the bound;

$$t-\pi[t+1,1;v] \geq \left(\frac{k^2+2k+1}{k^2+2k+2}\right)\left(\frac{v}{k+1}\right)-(2k^3-5k^2+6k-1).$$

An interesting corollary of Theorem 5.1 is

Corollary 5.2 [L10]: $3-\pi[4,1;v] \geq (v-2)/5$ for all $v \geq 172$.

A similar corollary was established for STS with a few small possible exceptions which were recently settled by Lo Faro to establish:

Theorem 5.3 [L10, L13]: $2-\pi[3,1;v] \geq (v-1)/4$ for all $v \geq 9$.

Brouwer has recently established an asymptotic bound for STS regarding the size of the maximum PPC. Given a STS(v), let π be a maximum PPC. Hence, there are $r = v-3|\pi|$ elements which do not occur in this PPC. We wish to bound r.

Theorem 5.4 [B17]: Given a STS(v), $r<5v^{2/3}$, where r is the number of elements not contained in a maximum PPC.

Brouwer also establishes the same asymptotic bound $r=O(v^{2/3})$ for SQS. We do not include the proofs here, although they are algorithmic in nature.

So far, we have examined only lower bounds on the size of a maximum PPC. However, what is the upper bound on $t-\pi[k,\lambda;v]$? This question has been posed by Phelps [P8] and probably others; can one construct STS(v), for v sufficiently large such that the STS does not contain a PPC with more than $(v-c)/3$ blocks for some $c \geq 4$? At the moment, it is not known whether there exists an STS(v) without a PC for each admissible value of v.

Other bounds have been established for the more general case of partial triple systems (PTS). A PTS is a simple 3-uniform hypergraph; *simple* means that every pair is contained in at most one triple. Consider the case where the PTS is maximal; in other words, the addition of any triple will cause some pair to occur in more than one triple. If one is trying to determine the size of the maximum PPC in a maximal PTS, this is equivalent to finding the size of the largest maximal matching in a maximal 3-uniform hypergraph. Working in terms of hypergraphs, Phelps [P8] established that

Theorem 5.5 [P8]: Every maximal simple 3-uniform hypergraph with n vertices contains a matching of size $n/12$.

Other researchers have examined the enumeration of resolvable designs. For example, the following enumerations have determined, in addition to other properties, which of the generated designs are resolvable: B[3,2;9] designs [M6], B[3,2;10] designs with repeated blocks [G1, G2], STS(21) with particular automorphism groups [M5]. In the case of MTS (for example, see [G1, G2]), the number of resolutions is simply the number of distinct resolutions of the underlying B[3,2;v] times 2^x, where x is the number of repeated blocks in the design. It appears that the problem has not been examined for related class of DTS.

A related study [M7] concerns the enumeration of 1-factorizations of the complete 3-uniform 9-vertex hypergraph; to restrict the task to a manageable size, only 1-factorizations with automorphism groups of size greater than 4 were considered. This study also includes an examination of some indecomposable and resolvable NB[3,λ;v] designs (i.e. no repeated blocks). Consider a resolvable B[3,λ;v]; it is said to be *$R-$decomposable* if at least one of its resolutions contains a resolution of a subdesign B[3,λ';v] with $0<\lambda'<\lambda$; otherwise the design is *$R-$indecomposable*. It is well-known that any resolvable B[3,2;9] is R-decomposable [M6]; this is not true for B[3,3;9] [M7]. In the case of B[3,3;9], any R-indecomposable design is also indecomposable [M7]; this is not the case for $\lambda > 3$.

Another enumeration study, that of cyclic SQS(20), is of interest here because Phelps [P7] recasts the enumeration problem in terms of perfect matchings in hypergraphs. Consider a cyclic SQS(v); associated with each quadruple (i,j,k,l) is a difference quadruple $<j-i,k-j,l-k,i-l>$, where the differences are taken mod v. The four 3-element subsets of a quadruple will give rise to four (not necessarily distinct) difference triples; these in turn characterize the orbit of the quadruple. Two quadruples will be in the same orbit (e.g. generated from the same base block) if and only if they have equivalent difference triples; two difference triples are *equivalent* if they are the same up to a cyclic reordering. Hence, there are three ways of characterizing the orbits of a cyclic SQS(v): one can choose a quadruple from each orbit or associate with each orbit a difference quadruple or a set of difference triples.

If one considers the difference triples to be the vertices of a hypergraph, the edges then correspond to the different orbits. Then a cyclic SQS(v) is equivalent to a 1-factor or perfect matching in this hypergraph. Using this setting, Phelps [P7] enumerated all cyclic SQS(20).

First consider all valid orbits of quadruples of Z_{20}. Let X be the set of all possible difference triples mod 20. Then to each valid orbit, assign the appropriate subset of X. Let E denote the collection of these subsets of X; hence, (E,X) is a hypergraph. If one locates a 1-factor in (E,X), then one has determined a set of edges such that every difference triple occurs exactly once in this set. Hence, the union of the corresponding orbits will be a cyclic SQS(20) since every triple will occur exactly once in this set of quadruples.

A brute-force search to locate 1-factors in (E,X) is, of course, undesirable. Instead, one employs the automorphism group to restrict the search. For example, having found all 1-factors that contain a particular edge, one can eliminate that edge and all edges in its orbit. Similarly, having found a particular partial 1-factor, the subgroup of these automorphisms that fixes this partial 1-factor can be employed to simplify the search.

The search can also be restricted by insisting upon the inclusion of a particular difference quadruple and hence set of difference triples. This is the case with cyclic SQS(20) which all contain a particular base block which corresponds to the difference triple (5,5,10). Therefore, this difference triple can be removed from the hypergraph along with all edges incident to it. Working in this reduced hypergraph, Phelps located all 1-factors via a simple backtracking algorithm; for details see [P7]. For other applications of this approach, again concerning cyclic SQS, see [C19, C28].

One interesting question regarding resolvable combinatorial configurations concerns orthogonal resolutions. For example, given a block design along with a particular resolution, can one determine a second resolution which is orthogonal to the first? In particular, a Kirkman system $B[k,1;v]$ is *doubly resolvable* if the blocks can be resolved into two resolutions R_1 and R_2 such that any resolution class from R_1 has at most one block in common with any resolution class from R_2. Similarly, doubly resolvable nearly Kirkman systems can be defined.

Room squares are examples of doubly resolvable Kirkman systems with $k=2$. Many researchers have examined Room squares; for example, see [M18, M19, R8, R9, S3]. Mathon and Vanstone [M8] constructed the first examples of *doubly resolvable* Kirkman systems with $k \geq 3$; in fact, they established the existence of infinite families of such designs. There are now a variety of construction techniques for doubly resolvable Kirkman systems and related combinatorial configurations; for example, see [D7, F2, F3, M8, V2]. Another generalization of Room squares, which has recently been examined, allows for each symbol to appear u times in each row and column of the array [L2, L3]. Furthermore, the construction of doubly resolvable designs has been greatly facilitated by a related combinatorial object called a *frame* [C27, M17].

Another resolvability problem worthy of note is that of resolving complete block designs. Baranyai [B4, C1] established that if $h \mid n$ then the h-element subsets of an n-element set can be partitioned into $\binom{n-1}{h-1}$ classes so that every class contains n/h disjoint h-element sets and every h-element set appears in exactly one class. In other words, if $k \mid v$, a complete block design can be resolved.

6. Decomposing Block Designs

Considering a t-$B[k,\lambda;v]$ design, one means of constructing such designs is to take the union of a t-$B[k, \lambda_1; v]$ design and a t-$B[k, \lambda_2; v]$ design where $\lambda = \lambda_1 + \lambda_2$. An obvious question is whether there are t-$B[k,\lambda;v]$ designs which cannot be expressed in this way. Such systems are referred to as *indecomposable* or *irreducible* designs; for a recent survey, see [S9]. Kramer [K7] demonstrated the existence of indecomposable $B[3,2;v]$ and $B[3,3;v]$ designs. Moreover, he showed that for $\lambda=2$ determining whether a design is decomposable can be carried out efficiently, i.e. in polynomial time. To do this, one constructs a block intersection graph in which adjacency of blocks denotes a shared pair of elements. This graph is bipartite if and only if the $B[k,2;v]$ is decomposable. Determining whether a graph is bipartite can be done in linear time [A1]. Kramer [K7] also observes that "the determination of indecomposability appears generally to be a difficult problem".

In [C14, C18], it is proved that deciding whether a B[3,3;v] is decomposable is NP-complete, and hence unlikely to have any efficient solution. NP-completeness is established by reducing the completion problem for commutative Latin squares, which has recently been shown to be NP-complete [C6, C8], to decomposability. We follow the proof given in [C14].

Given a r-regular n-vertex graph G, a ***Latin background*** for G, denoted LB[G;m,s] is an s x s symmetric array with elements chosen from $\{1,2,...m\}$. Each diagonal entry contains the element m. In the first n rows, each position is either empty, or contains a single element from the set $\{r+1,...,m\}$. In the latter s-n rows, each position contains a single element of the set $\{1,2,...m\}$. Each element appears at most once in each row (and symmetrically, each column). Finally, the pattern of empty squares forms an adjacency matrix for the graph G -- hence the term ***background***.

In [C6, C8], Cruse's embedding technique for partial commutative Latin squares [C41] is adapted to show that

Theorem 6.1 [C6, C8]: For each $r \geq 0$ and each r-regular n-vertex graph G, there is a Latin background LB[G;m,m] for every even $m \geq 2n$. Furthermore, one can be produced in time bounded by a polynomial in m.

Latin backgrounds are partial commutative Latin squares. In fact, a Latin background for a r-regular graph G can be completed (with no additional rows and columns) to a Latin square if and only if G is r-edge-colourable. Holyer [H12] has shown that deciding whether an arbitrary cubic graph is 3-edge-colourable is NP-complete, and Leven and Galil [L4] have generalized this result to r-edge-colourability for all fixed $r \geq 3$. Latin backgrounds are used to translate these graph theoretic results into the domain of combinatorial design theory.

Theorem 6.2 [C14]: Deciding whether an NB[3,λ;v] design contains a B[3,1;v] design is NP-complete.

Proof:

Membership in NP is immediate -- a nondeterministically chosen sub-B[3,1;v] can easily be verified in polynomial time. To show completeness, we reduce the known NP-complete problem of r-edge-colourability of r-regular graphs to our problem. Given an arbitrary n-vertex r-regular graph G, we first determine a size for a Latin background for G. When $2n-1 \equiv 1,3 (mod 6)$, we set $v = 2n-1$; otherwise we set $v = 2n+1$. Using theorem 6.1, we next construct a Latin background LB[G;$v+1$,$v+1$] called L; we do this in polynomial time. We produce r disjoint Latin backgrounds $L_1,...,L_r$ by repeatedly applying the permutation $(1\ 2\ ...r)\ (r+1,...v)\ (v+1)$ to the elements of L.

Using these Latin backgrounds, we construct an NB[3,r;2v+1] block design with elements $\{x_1, ..., x_v, y_1, ..., y_{v+1}\}$. The blocks are as follows:

(1) On the elements $\{x_1, ..., x_v\}$, place r disjoint Steiner triple systems. Such systems exist (at least) for all $v > 12r$, $v \equiv 1,3 \pmod 6$.
(2) Let $1 \le i < j \le v+1$ and let the (i,j) entry of one of the Latin backgrounds be k. Include the block $\{x_k, y_i, y_j\}$.
(3) Let $1 \le i < j \le v+1$ and let the (i,j) entry of the Latin background L be empty. Include the blocks $\{x_1, y_i, y_j\}$, $\{x_2, y_i, y_j\}$, and $\{x_3, y_i, y_j\}$ each once.

That the set of triples so defined forms an NB[3,r;2v+1] is easily verified, and this design is constructed in polynomial time. To establish NP-completeness, then, we need only show that the block design D has a sub-B[3,1;2v+1] if and only if G is r-edge-colourable; further, this depends only on the triples of type (3) above.

Suppose we have an r-edge-colouring of G. To find a sub-B[3,1;2v+1], we include the triples $\{\{x_k, y_i, y_j\} \mid \{y_i, y_j\}$ has colour $k\}$. Together with one of each of the disjoint Steiner triple systems, and the triples arising from one of the disjoint Latin backgrounds, this constructs a B[3,1;2v+1].

In the other direction, suppose D has a B[3,1;2v+1]. In this B[3,1;2v+1], the pairs appearing with x_1 (similarly, with x_2 and so on) form a 1-factor of G. Moreover, these 1-factors are all disjoint, and hence cover all edges of G. Since there are r disjoint 1-factors, and $(rv)/2$ edges in a r-regular graph, the 1-factors comprise a r-edge-colouring of G, as required.

Theorem 6.2 shows that deciding decomposability of NB[3,3;v] designs is NP-complete. However, it does not establish this for any $\lambda \ge 4$. The theorem can be generalized to establish that deciding whether a B[3,λ;v] design contains a B[3,1;v] design is NP-complete. Now consider the case $\lambda = 4$; an NB[3,4;v] may decompose into two NB[3,2;v] designs. Therefore, theorem 6.2 does not establish that determining whether a B[3,4;v] design is decomposable is NP-complete; although this is the case [C14].

Note that a design NB[3,λ;v] constructed by the process in Theorem 6.2 contains a NB[3,λ*prime*;v] if and only if the original λ-regular graph contains a λ'-factor. Konig [K6] has shown that whenever λ is odd, there are λ-regular graphs containing no regular factors. Applying the construction in Theorem 6.2 to these graphs produces indecomposable NB[3,λ;v] designs for every odd λ. Together with the constructions in [C31], this yields many infinite families of indecomposable triple systems with arbitrary odd λ.

For even λ the construction does not have such immediate applicability. Petersen [P2] has shown that every regular graph of even degree can be edge-partitioned into 2-factors; thus, all designs with even λ produced by this construction will be decomposable.

Although decomposing block designs appears to be a difficult problem, perhaps its correspondence with graph edge-colouring problem can be exploited in order to develop a heuristic algorithm. Given a B[3,3;v] design, consider the pairs which appear with a given element x. These pairs form a multigraph. A decomposition of the original triple system into three STS produces a 3-colouring of the edges of the multigraph. Therefore, a necessary condition is that the multigraphs associated with each of the v elements must be 3-edge colourable. More generally, if a given B[3,λ;v] design to be decomposed into n designs with $\lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_n$, one must be able to partition each multigraph into a λ_1-factor, a λ_2-factor, ... and a λ_n-factor. It is important to note that this condition is not sufficient. Consider a B[3,4;10] in which each of the 10 multigraphs has 9 vertices of degree 4. Therefore, Petersen's Theorem [P2] guarantees the existence of a 2-factor in each of these multigraphs. Yet it is known that there exists a B[3,4;10] design which cannot be decomposed [C31].

7. Embedding and Completing Block Designs

Our intent here is not to provide a comprehensive survey of embedding results, although most embedding and completing results are constructive and hence algorithmic in nature. For an excellent survey concerning embedding results for Steiner systems, the reader should consult [L8].

First, it is important to note that there exist partial designs which cannot be *completed*; that is, there is no set of blocks which can be added to the partial design such that a design is created on the same element set. An obvious example is the set of triples {(1 2 3) (4 5 6)}; it is impossible to complete this set of triples to a STS since $6 \neq 1$ or 3 (mod 6).

Since not all partial designs can be completed, one might ask whether there exist partial designs which cannot be embedded. When one is *embedding* a partial design, one is allowed to increase the number of elements. In 1971, Treash [T3] proved that every partial STS can be finitely embedded in an STS; unfortunately, the containing system is exponentially larger than the initial partial system. However, this need not be the case. Lindner [L8] proved that a partial STS(v) can be embedded in an STS(w) for any $w \geq 6v+1$, and of course $w \equiv 1,3 \pmod 6$. Andersen, Hilton and Mendelsohn [A2] improved the bound to $w \geq 4v+1$. Finally, Lindner [L9] has conjectured that the result can be improved to $w \geq 2v+1$, which is the best possible (if true).

Of course, given a particular STS, it may be possible to embed it into a system with fewer than $2v+1$ elements; in particular, it may be possible to complete the system. One question one might ask is "Can one easily determine the minimal number of elements required to embed a particular STS?". The answer appears to be no. In fact, a good characterization of those partial STS having very small embeddings is quite unlikely.

Theorem 7.1 [C8]: Deciding whether a partial STS(v) can be embedded in an STS (w) for some $w \leq 2v-1$ is NP-complete.

Proof:

Membership in NP is immediate. To establish completeness, start with an arbitrary cubic n-vertex graph G, and construct both an $IB(G;2n-1,2n-1)$ and an $IB(G;2n+1,2n+1)$. From a Latin background $B(G;m,m)$, one can construct an idempotent Latin background $IB(G;m-1,m-1)$ by placing the elements of the last row (column) along the diagonal, thereby eliminating the last row, the last column, and the last element. Then, simultaneously, interchange pairs of rows and pairs of columns, to place i in square (i,i). Recall from Theorem 6.1 that one can construct the appropriate Latin backgrounds in polynomial time and, therefore, the desired idempotent Latin backgrounds.

Either $2n+1$ or $2n-1$ is the order of a STS; let v denote which one is. Then construct a partial STS($2v+1$) with elements $\{x_1,\ldots,x_v,y_1,\ldots,y_v,z\}$. On the $\{x_i\}$ place the blocks of a STS(v). Next include the blocks $\{(z,x_i,y_i) \mid 1 \leq i \leq v\}$. Finally, whenever the (i,j) position of the $IB(G;v,v)$ is not empty, but rather contains an element k, we include the block (y_i,y_j,x_k). Since $IB(G;v,v)$ is idempotent, $k \neq i$ and $k \neq j$.

This partial system S can be completed if and only if G is 3-edge-colourable. Moreover, if S cannot be completed, at least one additional element e not in S must be added to complete it. This element must appear in triples with each element of S. In particular, e appears in triples with each element of $\{x_1,\ldots,x_v,z\}$. Each such triple requires a new element, not in S, since all pairs involving such an element with another element of S are already covered. Thus, the completion of S requires at least $v+2$ additional elements. This is not all, however.

In a completion of S, each of the $v+2$ additional elements may appear with some "edges" of the cubic graph G. Consider such an additional element which appears with the fewest edges of G. Here G has $3n/2$ edges, where $n \leq (v+1)/2$. There are at least $v+2$ additional elements. Thus, some element f appears with no edges of G. Every triple containing f involves at most one element of S. Thus, at least $2v+2$ additional elements (including f) are required. Hence, S cannot be completed in fewer than $4v+3$ elements.

Instead of examining the embedding of partial STS, one might wish to know whether a particular STS is contained in a larger design as a subdesign. If the containing design has the same number of elements (but larger λ), one is asking whether the larger design is decomposable. If, however, λ is kept constant, one is asking an embedding question. For any STS (V,B), and contained subdesign (V',B'), $|V| \geq 2|V'|+1$. However, given an arbitrary STS (V,B) can one guarantee that there exists a STS$(2|V|+1)$ which contains (V,B) as a subsystem? The answer is yes.

Theorem 7.2 [D10]: Any STS(v) can always be embedded in a STS(u) for every $u \geq 2v+1$.

If one does not restrict λ to 1, there are several appropriate embedding results for triple systems. For partial triple systems, finite embeddings are known for every λ [L11]. In the case of even λ, embeddings exist for which the size of the containing system is quadratic in the size of the partial system [C24]; for $\lambda=2$, this can be improved to linear [H4]. Both results rely on the fact that every triple system with even λ can be transformed into a DTS [C26]. Avoiding this preprocessing, linear embeddings can be obtained for triple systems with any λ [C23].

Many embedding results for block designs depend on embedding techniques for Latin squares and similar algebraic configurations. Therefore, of particular note is the fact that completing (not necessarily commutative) partial Latin squares is NP-complete [C9].

8. Orienting and Directing Block Designs

8.1 Orienting Block Designs

So far we have only examined designs in which the blocks are unordered k-subsets. Instead, one could assign an ordering to each block and require that each ordered t-tuple appear exactly λ times. For example, a *Mendelsohn triple system* of order v, denoted MTS(v), is a collection of triples such that every ordered pair appears exactly once. Each triple $(a\ b\ c)$ is considered to contain the ordered pairs $(a\ b)$ $(b\ c)$ and $(c\ a)$. Given a MTS(v), ignoring the ordering produces a B[3,2;v]. The converse is not necessarily true; for each admissible v, there exists a B[3,2;v] design which cannot be *oriented* to produce a MTS(v) [B8]. In fact, one can efficiently determine whether a B[3,2;v] design can be oriented; the algorithm is implicit in Mendelsohn's initial paper concerning these designs [M11].

The algorithm can also be presented in terms of 2-CNF Satisfiability, a problem which is well-known to computer scientists. The problem is as follows: one is given a set of variables U and a set of clauses over U. Each clause is the disjunction of two literals; a literal is either a variable or its negation. Hence, a clause is satisfied by a truth assignment if and only if at least one of its members is true under that assignment. The collection C of clauses is satisfiable if there exists some truth assignment for U that simultaneously satisfies all the clauses in C.

Even, Itai and Shamir [E4] proved that given such a collection C of clauses, one can in linear time determine whether there exists a satisfying truth assignment for C. If C is indeed satisfiable, the algorithm will produce a satisfying truth assignment.

Now consider a B[3,2;v] design. A block $(a\ b\ c)$ can be oriented in two distinct ways: $(a\ b\ c)$ which contains the ordered pairs $(a\ b)$, $(b\ c)$ $(c\ a)$ and $(a\ c\ b)$ with the ordered pairs $(a\ c)$ $(c\ b)$ $(b\ a)$. Arbitrarily orient each block of the design. Each block is then assigned a distinct label; these labels form the set of variables U. If the ordered block $(a\ b\ c)$ is assigned the label X, the alternate orientation $(a\ c\ b)$ is associated with the literal $\overline{X}$. A $v \times v$ matrix is created in which the $(a\ b)$ entry is the label(s) of the block(s) containing the ordered pair $(a\ b)$. This data structure is now employed in forming the clauses. For each pair $(a\ b)$, two clauses are created. Consider the case where the ordered pair $(a\ b)$ is contained in two blocks X and Y. Clearly, in the oriented design, only one of these blocks can contain the pair; the other block must be reordered. Therefore, one wants to satisfy the two clauses (X or Y) and ($\overline{X}$ or $\overline{Y}$) (i.e. the exclusive or of the two literals X and Y). Alternatively, if $(a\ b)$ is contained in block X and $(b\ a)$ is contained in Y, the two clauses are (X or $\overline{Y}$) and ($\overline{X}$ or Y) (i.e., $(a\ b)$ is contained in $\overline{Y}$).

One has now created a collection C of clauses. A satisfying truth assignment for C will specify the orientation of each block. Evan, Itai and Shamir's result proves that a satisfying assignment can be found in time linear in the number of clauses. Because we created these clauses in time linear in the number of blocks of the design, the result is a very efficient algorithm to determine whether a B[3,2;v] can be oriented to produce a MTS(v).

Unfortunately, an efficient algorithm is not known when $k \geq 3$. Given a B[k,2;v], one would like to orient each block such that each ordered pair appears exactly once; the ordered block $(x_1, x_2, \cdots x_k)$ is considered to contain the ordered pairs (x_i, x_j), where $i < j$ with one exception: the pair (x_k, x_1) is included instead of the pair (x_1, x_k). Deciding whether a B[k,2;v] design can be oriented appears to be a difficult problem, although it has not been shown to be NP-complete.

8.2 Directing Block Designs

Mendelsohn designs represent one means of interpreting ordered blocks; ***directed*** designs are another. Again consider a B[3,2;v] design. However, this time the ordered block (a b c) contains the ordered pairs (a b), (a c) and (b c). With this in mind, a ***directed triple system*** of order v, denoted DTS(v), is defined to be a collection of ordered triples such that each ordered pair appears exactly once; these systems are also called ***transitive triple systems***. Again one might ask which B[3,2;v] designs can be directed to produce DTS(v)? The answer is all of them [C16]. Moreover, C.Colbourn and Harms have extended the result to higher λ; in fact, they have demonstrated, the existence of a linear time algorithm for producing a directed design DB[3,λ;v] from a B[3,2λ;v] design [C26, H7, H8].

We present here a description of the algorithm for directing a triple system by illustrating its application to a particular example, a B[3,2;9] design. First one sorts the blocks into order:

013 015 024 028 035 046 067 078 127 127 135 146 148 168 234 236 256 258 348 367 378 457 457 568.

Next the design is partitioned into segments; a *segment* S_i is the collection of blocks having i as their first element. These can be easily identified from the sorted list of blocks:

S_0: 013 015 024 028 035 046 067 078
S_1: 127 127 135 146 148 168
S_2: 234 236 256 258
S_3: 348 367 378
S_4: 457 457
S_5: 568

For each segment S_i, we produce a *segment graph* G_i which contains the unordered pairs appearing with i in a triple of S_i. In our example, the segment graphs have the following edge sets:

G_0: 13 15 24 28 35 46 67 78
G_1: 27 27 35 46 48 68
G_2: 34 36 56 58
G_3: 48 67 78
G_4: 57 57
G_5: 68.

Segment graphs may be connected (as G_2 is) or disconnected (as G_1 is). In the event that the segment graph is disconnected, we define a ***subsegment*** to be a collection of blocks corresponding to a connected component of the segment graph. Each segment S_i can be partitioned into subsegments $S_{i,1}$, ..., $S_{i,m}$; each subsegment has a connected ***subsegment graph***. In our example, the subsegment graphs $G_{i,j}$ are:

$G_{0,1}$: 13 15 35
$G_{0,2}$: 24 28 46 67 78
$G_{1,1}$: 46 68 48
$G_{1,2}$: 35
$G_{1,3}$: 27 27
$G_{2,1}$: 34 36 56 58
$G_{3,1}$: 48 67 78
$G_{4,1}$: 57 57
$G_{5,1}$: 68.

These subsegment graphs can easily be produced in time which is linear in the size of the design. For each subsegment graph $S_{i,j}$, we locate all vertices of odd degree and add a 1-factor of ***virtual*** edges to construct an ***augmented subsegment graph*** $A_{i,j}$ in which every vertex has even degree. In our example, the virtual edges are as follows:

$A_{1,2}$: 35
$A_{2,1}$: 48
$A_{3,1}$: 46
$A_{5,1}$: 68.

We next examine the segments in reverse order, producing 3-tuples corresponding to the original set of blocks in such a way that no ordered pair ever appears more than once. The subsegments for each segment are handled in turn. In order to process a subsegment $S_{i,j}$, we first locate an Eulerian circuit in the augmented graph $A_{i,j}$. Each unordered pair appearing as an edge of $A_{i,j}$ corresponds either to a virtual edge or to an unordered pair in the design. In the case that $\{x,y\}$ appears in a block (i,x,y), we check whether the ordered pair (x,y) has already been employed once -- if not, we set $f(\{x,y\})=(x,y)$, and if so, we set $f(\{x,y\})=(y,x)$. This function f determines the order in which the elements $\{x,y\}$ will appear in the 3-tuple replacing the block (i,x,y).

Two cases arise, according to whether the length of the Eulerian circuit is even or odd. When the length of the Eulerian circuit in $A_{i,j}$ is even, we construct a set of 3-tuples from the edges of the augmented graph by processing the edges in order along the Eulerian circuit; the element i is alternately placed at the beginning and the end of a triple. For virtual edges, no triple is produced; for edges arising from blocks of the design, the ordering of the other two elements is prescribed by f. When the length of the Eulerian circuit is odd, one triple is chosen to have i placed in the middle, but otherwise the beginning/end alternation is followed as before.

We illustrate the application of this method on our example. For each subsegment, we have listed the edges of the Eulerian circuit in the augmented graph (in order) in the first column. The second column gives the value of the function f computed for non-virtual edges, and the third column gives the directed block produced:

Subsegment Graph	Eulerian Circuit	Value of f	Block Included
$G_{5,1}$	68	68	568
	86 (virt)		
$G_{4,1}$	57	57	457
	75	75	754
$G_{3,1}$	48	48	348
	87	87	873
	76	76	376
	64 (virt)		
$G_{2,1}$	56	65	625
	63	63	263
	34	43	432
	48 (virt)		
	85	85	852
$G_{1,1}$	68	86	816
	84	84	184
	46	46	461
$G_{1,2}$	35	35	135
	53 (virt)		
$G_{1,3}$	27	27	127
	72	72	721
$G_{0,1}$	13	31	301
	35	53	053
	51	51	510
$G_{0,2}$	24	24	204
	46	64	640
	67	67	067
	78	78	780
	82	28	028

Eulerian circuits can be found quickly, and it is a simple matter to keep track of the pairs already used in order to evaluate the function f. Thus the entire directing algorithm requires only a linear amount of time in the size of its input.

In the case $k=3$, one is attempting to direct each block of the design into a transitive tournament of order 3. From the above results, we know this can always be accomplished. One might ask if this extends to transitive tournaments of higher orders. It remains an open question whether there exists a $(v,k,2)$ design which cannot be directed into the transitive tournament of k vertices. Either a general algorithm or a counterexample would be of interest here.

In the case $k=3$, Harms has also examined cyclic systems; an efficient algorithm for producing a directed cyclic triple system from an undirected one is presented elsewhere in this volume [H6].

Other algorithmic questions concerning directing have been posed by Teirlinck. Given an idempotent commutative quasigroup, consider the upper triangle. Let the (i,j) entry be k; this can be viewed as the block (i,j,k). Therefore, the upper triangle of an idempotent commutative quasigroup corresponds to a triple system with $\lambda=3$. Now consider the converse. Given a triple system with $\lambda=3$, can it be written as the upper triangle of an idempotent commutative quasigroup? One can ask a similar question in the case when $\lambda=6$; given a triple system with $\lambda=6$, does it have a corresponding idempotent quasigroup? One can view these questions in terms of block orderings, for example, the blocks from the idempotent commutative quasigroup are "almost" ordered (the i and j can however be flipped). The answer to both questions is no. For example, consider the case with $\lambda=6$. Take a $(v,3,2)$ design which is not a MTS. Take each block three times; hence $\lambda=6$, but the design does not correspond to an idempotent quasigroup. However, one interesting question which remains is "What is the computational complexity of deciding, given the triple system, whether or not there exists a corresponding idempotent commutative quasigroup?"

9. Algorithmic Aspects of Intersection Problems

As noted in an earlier section concerning isomorphism invariants, instead of examining the design itself, one might choose to analyze the corresponding intersection graph. Although intersection graphs of designs have not been studied extensively for all families of designs, there are certain cases in which it is known that the intersection graphs possess certain characteristics. For example, if the blocks of a BIBD intersect in only two possible sizes, the corresponding block intersection graph is strongly regular [G9]; in particular, the intersection graphs of Steiner 2-designs are strongly regular graphs. In the case of twofold triple systems, each component of the pair intersection graph is cubic and 3-connected [C35]. A very nice series of results concerning intersection graphs is due to Poljak, Rödl and Turzik [P12], who examine the problem in terms of sets of distinct representatives for graphs.

A family $F=(A_x \mid x \in V)$ of sets, which are not necessarily distinct, is called a *set representation* of a graph $G=(V,E)$ if $A_x \cap A_y \neq \emptyset$ if and only if $(x,y) \in E$ for every pair x,y of distinct vertices of G. Conversely, G is called an *intersection graph* of F. A set representation F of G is called a *$k-$set representation* if $|A_x| \leq k$ for all $x \in V$ and a *distinct set representation* if $A_x \neq A_y$ for all $x,y \in V, x \neq y$. It is a *simple set representation* if $|A_x \cap A_y| \leq 1$ for all $x,y \in V, x \neq y$. It is well-known that every graph has a simple set representation [M1].

Poljak, Rödl and Turzik [P12] prove the following theorems:

Theorem 9.1 [P12]: It is NP-complete to find a minimum integer k for which a given graph G has a k-set representation.

Theorem 9.2 [P12]: It is NP-complete to decide whether a given graph G has a 4-set representation.

Theorem 9.3 [P12]: It is NP-complete to decide whether a graph has a distinct 3-set representation.

These results indicate that the characterization of *line graphs*, which are intersection graphs of graphs, probably cannot be generalized even for triples; one can determine in polynomial time whether a given graph has a (simple) 2-set representation [B7, B9]. Line graphs are characterized by a finite family of minimal forbidden induced subgraphs [B7]. However, for the graphs which are intersection graphs of k-hypergraphs, $k>2$, the analogous statement does not hold [P12].

Poljak, Rödl and Turzik also establish that

Theorem 9.4 [P12]: It is NP-complete to find the minimum k such that for a given graph G there exists a simple set representation with $|\cup F| = k$.

This result can also be considered in connection with line graphs, because if G is a graph and H is the line graph of G, then G is a simple set representation of H.

From the above results, it appears that it is hard to decide whether a graph is the intersection graph of a design. As intersection graphs of hypergraphs cannot be easily characterized, it seems unlikely that one will be able to characterize intersection graphs of designs, although this may be possible for restricted families of designs.

Instead of looking at just the intersection graph of one design, one can examine the intersection patterns or graphs of two or more designs. The family of designs which has received the most attention in this regard is Steiner systems, particularly STS. Common questions which have been posed include [R10]:

> Given two Steiner $S(t,k,v)$ systems on the same v-set, how many blocks in common can they have?
> Can one find two such systems with no blocks in common?
> If yes, what is the largest number of such systems such that no pair of systems have a block in common?

Two designs (V,B_1) and (V,B_2) are *disjoint* if $B_1 \cup B_2 = \emptyset$ i.e. they have no blocks in common. Many researchers have examined the existence of disjoint Steiner systems and, in particular, have tried to determine the maximum number of pairwise disjoint Steiner systems. For an excellent survey of research concerning intersection patterns of Steiner systems, the reader should consult [R10].

Of particular interest here are some of the algorithmic results concerning intersection patterns.

Theorem 9.5 [L7, T1]: If (V_1,B_1), (V_2,B_2) are any two $S(2,3,v)$ systems, $v \geq 7$, and if V is any v-set, then there exists two disjoint $S(2,3,v)$ systems (V,B_1), (V,B'_2) such that (V,B_2) is isomorphic to (V,B'_2).

We include Lindner's version of the proof here:

Proof:

Let (V,B_1) and (V,B_2) be any two Steiner triple systems of order v. Let $(1,2,3)$ be any triple in $B_1 \cap B_2$ and define the *spread* of 3, denoted by $s(3)$, to be

$$s(3) = (1,2,3) \cup A \cup C, \text{ where}$$
$$A = \{a \mid (z,w,a) \in B_1 - \{(1,2,3)\} \text{ and } (z,w,3) \in B_2\}, \text{ and}$$
$$C = \{b \mid (x,y,b) \in B_2 - \{(1,2,3)\} \text{ and } (x,y,3) \in B_1\}.$$

With these definitions in mind, the following two statements can easily be verified.

(1) If $|s(3)| < v$ and $d \in V - s(3)$, then $|B_1 \cap B_2| > |B_1 \cap B_2(3d)|$, where $B_2(3d)$ is the collection of triples obtained by interchanging 3 and d in the triples of B_2.

(2) If $|s(3)|=v$ and d is any point in $A \cup C$, then $|B_1 \cap B_2| = |B_1 \cap B_2(3d)|$.

Let (V,B_1) and (V,B_2) be any two triple systems such that $B_1 \cap B_2 \neq \emptyset$. One of two things is true: either there is a triple in $B_1 \cap B_2$ containing a point whose spread is less than v or there is no such triple. Because we have two possible cases, we introduce two distinct procedures which one can continue to apply until (V,B_1) and (V,B_2) are disjoint. Whenever there is a triple in $B_1 \cap B_2$ which contains a point whose spread is less than v, Procedure 1 is applied, else Procedure 2 can be used. After employing Procedure 2, one is guaranteed that Procedure 1 is applicable.

Procedure 1: Let $(1,2,3) \in B_1 \cap B_2$ and $|s(3)| < v$. Choose $d \in V - s(3)$. Now interchange elements 3 and d in (V,B_2). We know that $|B_1 \cap B_2| > |B_1 \cap B_2(3d)|$, and of course (V,B_2) is isomorphic to $(V,B_2(3d))$.

Procedure 2: From (V,B_2), select any triple containing 3 (other than the triple $(1,2,3)$); let this triple be $(3,x,y)$. In (V,B_1), the triple containing x and y cannot intersect $(1,2,3)$ since $|s(3)|=v$; let this triple be (c,x,y). Now consider the unique triple in (V,B_2) which contains 3 and c; let this triple be $(3,c,e)$. Now return to (V,B_1) and examine the triple containing c and e. Let this triple be (c,d,e); again, it cannot intersect $(1,2,3)$ since $|s(3)|=v$.

At this point, the triples we are examining in (V,B_1) are $(1,2,3)$, (c,x,y) and (c,d,e); we have selected three triples in (V,B_2) which are $(1,2,3)$, $(3,x,y)$ and $(3,c,e)$. Note that we have not examined the element d in (V,B_2). We can at this point interchange elements 3 and d in (V,B_2). We have not changed the number of blocks in which the two STS intersect; $|B_1 \cap B_2| = |B_1 \cap B_2(3d)|$. However, the triple $(c,d,e) \in B_1 \cap B_2(3d)$ and $|s(d)| < v$ (also $|s(c)| < v$). Hence, (V,B_1) and $(V,B_2(3d))$ have a triple in common which contains a point whose spread is less than v.

Given two STS (V,B_1) and (V,B_2), one can make them disjoint by repeatedly applying Procedure 1 whenever it is applicable, otherwise apply Procedure 2; Procedure 2 guarantees that Procedure 1 can then be employed again.

This algorithm can, of course, be employed to produce pairs of isomorphic disjoint STS; one simply starts with two copies of the same design.

A general result which is analogous to Theorem 9.5 is due to Ganter, Pelikán and Teirlinck:

Theorem 9.6 [G3]: If (V_1,B_1), (V_2,B_2) are any two $S(t,k,v)$ systems with $2t \le k < v$, then there exist two disjoint $S(t,k,v)$ systems (V,B'_1), (V,B'_2) such that (V_1,B_1) is isomorphic to (V,B'_1) and (V_2,B_2) is isomorphic to (V,B'_2).

In the case of SQS, Gionfriddo and Lindner [G6, G7, G8] have also constructed pairs of designs with prescribed intersection patterns. Their approach involves interchanging design fragments in order to change the number of blocks which two designs have in common.

10. Conclusions

As demonstrated throughout this paper, there remain many interesting open problems concerning various computational aspects of block designs. For example, the complexity of determining whether or not a design is resolvable is unknown. Although it is likely that the problem is NP-complete, this has not been established. Consider tasks such as determining whether a particular design can be nested or determining whether it is a derived design; the complexity of these operations is again unknown. A more general problem is determining the chromatic index of a design. Again this is an open problem, although the corresponding problem for graphs is known to be NP-complete [H12].

All of these problems are related. The operation of nesting requires that one increase both k and λ, while v and t remain fixed. When embedding a design, v is increased; t, k and λ are fixed. C. Colbourn, Hamm and Rosa [C25] examine a related operation in which v and λ are increased simultaneously. When determining whether a particular Steiner system is *derived* from another, one is increasing t, k and v by 1. For further information regarding derived Steiner systems, see [D6, G5, G10, M5, P6].

With regard to decomposing block designs, one might ask what is the smallest λ for which one can guarantee that a $B[k,\lambda;v]$ design can be decomposed [M4]. Still other computational problems concern orienting and directing designs; relatively little is known with regard to these operations on designs with $k>3$.

As mentioned in the introduction, many algorithmic issues concerning the construction of various combinatorial configurations have not been addressed here. Obviously, there remain many open questions regarding the existence of various families of block designs. Within this vein, one interesting computational result is the fact that determining whether a multiset of integers represents the block sizes of a PBD is NP-complete [C29]. A related open question, posed by Phelps [P9], is "What is the complexity of determining whether a multiset of integers is the degree sequence of a PBD?".

As demonstrated herein, there has been an extensive amount of research which is both computational and combinatorial in nature. Moreover, there are other algorithmic aspects and problems concerning various combinatorial configurations which we have not addressed here. The past interaction between combinatorics and computer science has benefitted both fields. Combinatorial tools have helped to produce efficient algorithms; for example, consider the polynomial-time algorithm for 2-colouring SQS. Moreover, computer science techniques have greatly aided in obtaining results in combinatorics. Hopefully, collaboration between the two fields will continue.

Acknowledgements

There are many people who have provided encouragement and invaluable assistance throughout the production of this paper. In particular, I want to thank Charlie Colbourn, Janelle Harms, Esther Lamken, Curt Lindner, Rudi Mathon, Kevin Phelps, Doug Stinson, Alex Rosa, and Scott Vanstone, for either reading earlier drafts of this manuscript, or providing references, proofs and helpful suggestions. Charlie and Alex deserve special thanks for reading the manuscript multiple times and for doing so in painstaking detail. Research of the author is supported by NSERC Canada under grant number A0764.

References

[A1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *Data Structures and Algorithms*, Addison-Wesley Publishing Co., 1983.

[A2] L.D. Andersen, A.J.W. Hilton and E. Mendelsohn, "Embedding Partial Steiner Triple Systems", *Proceedings London Math. Society, 41,* (1980) 557-576.

[A3] B.A. Anderson, "Sequencing and Houses", *Proceedings of the Fifteenth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1984, pp. 23-45.

[A4] E. Arjomandi, "An Efficient Algorithm for Colouring the Edges of a Graph with d+1 Colours", *INFOR, 20,* (1982) 82-101.

[B1] L. Babai, "On the Isomorphism Problem", unpublished manuscript, 1977.

[B2] L. Babai, P. Erdös and S.M. Selkow, "Random Graph Isomorphism", *SIAM J. Computing, 9,* (1980) 628-637.

[B3] L. Babai and E.M. Luks, "Canonical Labeling of Graphs", *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing,* 1983, 171-183.

[B4] Z. Baranyai, "The Edge-Coloring of Complete Hypergraphs I", *J. Combinatorial Theory (B), 26,* (1979) 276-294.

[B5] L.D. Baumert and S.W. Golomb, "Backtrack Programming", *JACM, 12,* (1965) 516-524.

[B6] S. Bays, "Sur les Systèmes Cycliques de Triples de Steiner Differents pour N Premier (ou Puissance de Nombre Premier) de la Forme $6n+1$" *Comment. Math. Helv., 2,* (1930) 294-305. II-III, *Comment. Math. Helv., 3,* (1931) 22-41. IV-V, *Comment. Math. Helv., 3,* (1931) 307-325.

[B7] L.W. Beineke, "Derived Graphs and Digraphs", *Beiträge zur GraphenTheorie,* Leipzig, (1968) 17-33.

[B8] F.E. Bennett and N.S. Mendelsohn, "Some Remarks on 2-Designs $S_2(2,3,v)$", *Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1979, 119-127.

[B9] J.C. Bermond and J.C. Meyer, "Graphs Representent des Arcs d'un Multigraphe", *J. Math. Pures et Appliques, Series 9, 52,* (1973) 229-308.

[B10] S. Bilaniuk and E. Mendelsohn, "A Survey of Coloring of Steiner Systems", *Proceedings of the Fifteenth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1984, pp. 127-140.

[B11] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications,* Basingstoke Macmillan, 1976.

[B12] K.S. Booth, "Lexicographically Least Circular Substrings", *Information Processing Letters, 10,* (1980) 240-242.

[B13] K.S. Booth and C.J. Colbourn, *Problems Polynomially Equivalent to Graph Isomorphism,* Technical Report CS-77/04, Dept. of Computer Science, University of Waterloo, 1979.

[B14] R.C. Bose, "On the Construction of Balanced Incomplete Block Designs", *Ann. Eugenics, 9,* (1939) 353-399.

[B15] R.C. Bose and S.S. Shrikhande, "Baer Subdesigns of Symmetric Balanced Incomplete Block Designs", in *Essays in Probability and Statistics,* S. Ikeda (ed), Shinko Tsusho Co., 1976, 1-16.

[B16]
N. Brand, "Isomorphic Designs That Are Not Multiplier Equivalent", preprint.

[B17]
A.E. Brouwer, "On the Size of a Maximum Transversal in a Steiner Triple System", *Canadian J. Math.*, *13*, (1981) 1202-1204.

[B18]
A. Bruen, "Blocking Sets in Finite Projective Planes", *SIAM J. Applied Math.*, *21*, (1971) 380-392.

[C1] P.J. Cameron, *Parallelisms of Complete Designs*, Lecture Notes Series, 23, London Math. Society, Cambridge University Press, 1976.

[C2] N. Christofides, *Graph Theory: An Algorithmic Approach*, Academic Press, New York, 1975.

[C3] A. Cobham, "The Intrinsic Computational Difficulty of Functions", in Y. Bar-Hillel (ed), *Proceedings 1964 International Congress for Logic Methodology and Philosophy of Science*, North Holland, Amsterdam, 1964, 24-30.

[C4] E.J. Cockayne, G.G. Miller and G. Prins, "An Interpolation Theorem for Partitions which are Complete with Respect to Hereditary Properties", *J. Combinatorial Theory (B)*, *13*, (1972) 290-297.

[C5] C.J. Colbourn, *A Bibliography of the Graph Isomorphism Problem*, Technical Report 123/78, Dept. of Computer Science, University of Toronto, 1978.

[C6] C.J. Colbourn, "Some NP-complete Problems on Graph Decompositions", *Proceedings of the Nineteenth Allerton Conference on Communications, Control, and Computing*, 1981, 741-745.

[C7] C.J. Colbourn, "Computing the Chromatic Index of Steiner Triple Systems", *The Computer Journal*, *25 (3)*, (1982) 338-339.

[C8] C.J. Colbourn, "Embedding Partial Steiner Triple Systems is NP-Complete", *J. Combinatorial Theory (A)*, *35*, (1983) 100-105.

[C9] C.J. Colbourn, "The Complexity of Completing Partial Latin Squares", *Discrete Applied Math.*, *8*, (1984) 25-30.

[C10]
C.J. Colbourn and K.S. Booth, "Linear Time Automorphism Algorithms for Trees, Interval Graphs, and Planar Graphs", *SIAM J. Computing*, *10*, (1981) 203-225.

[C11]
C.J. Colbourn and M.J. Colbourn, "Combinatorial Isomorphism Problems Involving 1-Factorizations", *Ars Combinatoria, 9,* (1980) 191-200.

[C12]
C.J. Colbourn and M.J. Colbourn, "Deciding Hadamard Equivalence of Hadamard Matrices", *BIT, 21,* (1981) 374-376.

[C13]
C.J. Colbourn and M.J. Colbourn, "The Chromatic Index of Cyclic Steiner 2-Designs", *International J. Math. and Math. Sci., 5 (4),* (1982) 823-825.

[C14]
C.J. Colbourn and M.J. Colbourn, "Decomposition of Block Designs: Computational Issues", *Proceedings of the Tenth Australian Conference on Combinatorial Mathematics,* Adelaide, Australia, 1982; *Lecture Notes in Mathematics, 1036,* Springer-Verlag, 1983, 141-146.

[C15]
C.J. Colbourn and M.J. Colbourn, "Greedy Colourings of Steiner Triple Systems", *Annals of Discrete Mathematics, 18,* (1983) 201-208.

[C16]
C.J. Colbourn and M.J. Colbourn, "Every Twofold Triple System Can be Directed", *J. Combinatorial Theory (A), 34,* (1983) 375-378.

[C17]
C.J. Colbourn and M.J. Colbourn, "Nested Triple Systems", *Ars Combinatoria, 16,* (1983) 27-34.

[C18]
C.J. Colbourn and M.J. Colbourn, "The Computational Complexity of Decomposing Block Designs", *Annals of Discrete Math.,* to appear.

[C19]
C.J. Colbourn, M.J. Colbourn and K.T. Phelps, "Combinatorial Algorithms for Generating Cyclic Steiner Quadruple Systems", *Proceedings of Discrete Mathematical Analysis and Combinatorial Computation,* Fredericton, 1980, 25-39.

[C20]
C.J. Colbourn, M.J. Colbourn, K.T. Phelps and V. Rödl, "Colouring Steiner Quadruple Systems", *Discrete Applied Math., 4,* (1982) 103-111.

[C21]
C.J. Colbourn, M.J. Colbourn, K.T. Phelps and V. Rödl, "Coloring Block Designs is NP-Complete", *SIAM J. Algebraic and Discrete Methods, 3 (3),* (1982) 305-307.

[C22]
C.J. Colbourn, M.J. Colbourn and W. Rosenbaum, "Trains: An Invariant for Steiner Triple Systems", *Ars Combinatoria, 13,* (1982) 149-162.

[C23]
C.J. Colbourn, R.C. Hamm, C.C. Lindner and C.A. Rodger, "Embedding Partial Graph Designs, and Triple Systems with $\lambda > 1$", Research Report CORR 84-14, Dept. of Combinatorics and Optimization, University of Waterloo, 1984.

[C24]
C.J. Colbourn, R.C. Hamm and C.A. Rodger, "Small Embeddings of Partial Directed Triple Systems and Partial Triple Systems with Even λ", *J. Combinatorial Theory (A),* 37, (1984) 363-369.

[C25]
C.J. Colbourn, R.C. Hamm and A. Rosa, "Embedding, Immersing and Enclosing", *Proceedings of the Sixteenth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1985, to appear.

[C26]
C.J. Colbourn and J.J. Harms, "Directing Triple Systems", *Ars Combinatoria, 15,* (1983) 261-266.

[C27]
C.J. Colbourn, K.E. Manson and W.D. Wallis, "Frames for Twofold Triple Systems", *Ars Combinatoria, 17,* (1984) 69-78.

[C28]
C.J. Colbourn and K.T. Phelps, "Three New Steiner Quadruple Systems", *Utilitas Math., 18,* (1980) 35-40.

[C29]
C.J. Colbourn, K.T. Phelps and V. Rödl, "Block Sizes in Pairwise Balanced Designs", *Canadian Math. Bulletin, 27,* (1984) 375-380.

[C30]
C.J. Colbourn and R.C. Read, "Orderly Algorithms for Graph Generation", *International J. Computer Math., 7,* (1979) 167-172.

[C31]
C.J. Colbourn and A. Rosa, "Indecomposable Triple Systems with $\lambda = 4$", *Studia Sci. Math. Hung.,* to appear.

[C32]
M.J. Colbourn, "An Analysis Technique for Steiner Triple Systems", *Proceedings of the Tenth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1979, 289-303.

[C33]
M.J. Colbourn, "The Analysis of Steiner Systems", *Proceedings of the Ninth Manitoba Conference on Numerical Mathematics and Computing,* Winnipeg, 1979, 159-164.

[C34]
M.J. Colbourn and C.J. Colbourn, "The Complexity of Deciding Isomorphism of Block Designs", *Discrete Applied Mathematics, 3,* (1981) 155-162.

[C35]
M.J. Colbourn and J.K. Johnstone, "Twofold Triple Systems with a Minimal Change Property", *Ars Combinatoria, 18,* (1984) 151-160.

[C36]
M.J. Colbourn and R.A. Mathon, "On Cyclic Steiner 2-Designs", *Annals of Discrete Mathematics, 7,* (1980) 215-253.

[C37]
F.N. Cole, "The Triad Systems of Thirteen Letters", *Transactions of American Math. Society, 14,* (1913) 1-5.

[C38]
S. Cook, "The Complexity of Theorem-Proving Procedures", *Proceedings of the 3rd Annual Symposium on Theory of Computing,* Association for Computing Machinery, New York, 1971, 151-158.

[C39]
S. Cook, "ACM Turing Award Lecture: An Overview of Computational Complexity", *Communications of the ACM, 26(6),* 1983, 401-408.

[C40]
D.G. Corneil, *Graph Isomorphism,* Technical Report 18, Dept. of Computer Science, University of Toronto, 1968.

[C41]
A.B. Cruse, "On Embedding Incomplete Symmetric Latin Squares", *J. Combinatorial Theory (A), 16,* (1974) 18-22.

[C42]
L.D. Cummings, "On a Method of Comparison for Triple Systems", *Transactions of American Math. Society, 15,* (1914) 311-327.

[D1] M. de Brandes, K.T. Phelps and V. Rödl, "Coloring Steiner Triple Systems", *SIAM J. Algebraic and Discrete Methods, 3,* (1982) 241-249.

[D2] M. de Brandes and V. Rödl, "Steiner Triple Systems with Small Maximal Independent Sets", *Ars Combinatoria, 17,* (1984) 15-19.

[D3] J. Dénes, "Connections Between Transformation Semigroups and Graphs", *Theorie des Graphes. Journee Internationales d'etude*, Gordon and Greach, 1967, 93-101.

[D4] J. Dénes, "On a Problem of A. Kotzig", *Annals of Discrete Math., 18,* (1983) 283-290.

[D5] M.J. de Resmini, "On Blocking Sets in Symmetric BIBD's with $\lambda > 2$", *J. Geometry, 18,* (1982) 194-198.

[D6] H.L. de Vries, I. Diener and E. Schmitt, "All 80 Steiner Triple Systems on 15 Elements are Derived", *Discrete Math.*, to appear.

[D7] L.J. Dickey and R. Fuji-Hara, "A Geometric Construction of Doubly Resolvable $(n^2+n+1,1)$-designs", *Ars Combinatoria, 8,* (1979) 3-12.

[D8] J.H. Dinitz and D.R. Stinson, "A Fast Algorithm for Finding Strong Starters", *SIAM J. Algebraic and Discrete Methods, 2 (1),* (1981) 50-56.

[D9] J. Doyen and M. Vandensavel, "Non Isomorphic Steiner Quadruple Systems", *Bulletin de la Societe Mathematique de Belgique, 23,* (1971) 393-410.

[D10] J. Doyen and R.M. Wilson, "Embeddings of Steiner Triple Systems", *Discrete Math., 5,* (1973) 229-239.

[D11] D.A. Drake, "Blocking Sets in Block Designs", preprint.

[E1] J. Edmonds, "Paths, Trees and Flowers", *Canadian J. Math., 17,* (1965) 449-467.

[E2] P. Erdös, "On the Combinatorial Problems which I Would Most Like to See Solved", *Combinatorica, 1,* (1981) 25-42.

[E3] S. Even, *Algorithmic Combinatorics,* Macmillan, New York, 1973.

[E4] S. Even, A. Itai and A. Shamir, "On the Complexity of Timetable and Multicommodity Flow Problems", *SIAM J. Computing, 5,* (1976) 691-703.

[F1] M. Fontet, *Connectivite des Graphes et Automorphismes des Cartes:* Proprietes et Algorithmes, Thesis; also report no. 79-23, Laboratoire Informatique Theorique et Programmation, CNRS, Universite Paris 7, 1979.

[F2] R. Fuji-Hara and S.A. Vanstone, "On the Spectrum of Doubly Resolvable Kirkman Systems", *Proceedings of the Eleventh Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1980, 399-407.

[F3] R. Fuji-Hara and S.A. Vanstone, "Transversal Designs and Doubly Resolvable Designs", *European J. Combinatorics, 1,* (1980) 219-223.

[G1] B. Ganter, A. Gülzow, R.A. Mathon and A. Rosa, "A Complete Census of (10,3,2) Block Designs and of Mendelsohn Triple Systems of Order 10: Part IV: (10,3,2) Designs with Repeated Blocks", Report 5/78, Mathematische Schriften, Gesamthochschule Kassel, Germany.

[G2] B. Ganter, R.A. Mathon and A. Rosa, "A Complete Census of (10,3,2) Block Designs and of Mendelsohn Triple Systems of Order 10: Part II: Mendelsohn Triple Systems with Repeated Blocks", *Proceedings of the Eighth Manitoba Conference on Numerical Mathematics and Computing,* 1978, 181-204.

[G3] B. Ganter, J. Pelikán and L. Teirlinck, "Small Sprawling Systems of Equicardinal Sets", *Ars Combinatoria, 4,* (1977) 133-142.

[G4] M.R. Garey and D.S. Johnson, *Computers and Intractability,* W.H. Freeman Publishing Co., San Francisco, 1979.

[G5] P.B. Gibbons, *Computing Techniques for the Construction and Analysis of Block Designs,* Ph.D. Thesis; also Technical Report 92, Dept. of Computer Science, University of Toronto, 1976.

[G6] M. Gionfriddo, "Some Results on Partial Steiner Quadruple Systems", *Annals of Discrete Mathematics, 18,* (1981) 401-408.

[G7] M. Gionfriddo, "On the Block Intersection Problem for Steiner Quadruple Systems", *Ars Combinatoria, 15,* (1983) 301-314.

[G8] M. Gionfriddo and C.C. Lindner, "Construction of Steiner Quadruple Systems Having a Prescribed Number of Blocks in Common", *Discrete Mathematics, 34,* (1981) 31-42.

[G9] J.M. Goethals and J.J. Seidel, "Strongly Regular Graphs Derived from Combinatorial Designs", *Canadian J. Math, 22,* (1970) 597-614.

[G10] M.J. Grannell and T.S. Griggs, "Derived Steiner Triple Systems of Order 15", elsewhere in this volume.

[G11] M. Guregová and A. Rosa, "Using the Computer to Investigate Cyclic Steiner Quadruple Systems", *Mat. Casopis, 18,* (1968) 229-239.

[H1] M. Hall, Jr., "Automorphisms of Steiner Triple Systems", *IBM J. Research and Development, 4,* (1960) 460-472.

[H2] M. Hall, Jr., *Combinatorial Theory,* Wiley Publishing Co., New York, 1967.

[H3] M. Hall, Jr. and J.D. Swift, "Determination of Steiner Triple Systems of Order 15", *Math. Tables Aids Comput., 52,* (1955) 146-152.

[H4] R.C. Hamm, C.C. Lindner and C.A. Rodger, "Linear Embeddings of Partial Directed Triple Systems with $\lambda = 1$ and Partial Triple Systems with $\lambda = 2$", *Ars Combinatoria, 16,* (1983) 11-16.

[H5] F. Harary and S. Hedetniemi, "The Achromatic Number of a Graph", *J. Combinatorial Theory, 8,* (1970) 154-161.

[H6] J.J. Harms, "Directing Cyclic Triple Systems", elsewhere in this volume.

[H7] J.J. Harms and C.J. Colbourn, "An Efficient Algorithm for Directing Triple Systems", *Proceedings of the Fourteenth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1983, 455-462.

[H8] J.J. Harms and C.J. Colbourn, "An Optimal Algorithm for Directing Triple Systems Using Eulerian Circuits", *Annals of Discrete Math.,* to appear.

[H9] A. Hartman, *Resolvable Designs,* M.Sc. Thesis, Israel Institute of Technology, Israel, 1978.

[H10] J. Hartmanis and R.E. Stearns, "On the Computational Complexity of Algorithms", *Transactions of American Mathematical Society, 117,* (1965) 285-306.

[H11] D.R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid,* Vintage Books, 1980.

[H12] I. Holyer, "The NP-completeness of Edge-colouring", *SIAM J. Computing, 10,* (1981) 718-720.

[H13] J.E. Hopcroft and R.E. Tarjan, "Isomorphism of Planar Graphs", in: R.E. Miller and J.W. Thatcher (eds), *Complexity of Computer Computations* Plenum Press, 1972, 131-151.

[H14] T.C. Hu, *Combinatorial Algorithms,* Addison-Wesley Publishing Co., 1982.

[I1] A.V. Ivanov, "Constructive Enumeration of Incidence Systems", elsewhere in this volume.

[K1] R.M. Karp, "Reducibility Among Combinatorial Problems", in R.E. Miller and J.W. Thatcher (eds), *Complexity of Computer Computations,* Plenum Press, 1972, 85-103.

[K2] R.M. Karp, "Probabilistic Analysis of a Canonical Numbering Algorithm for Graphs", *Proceedings AMS Symposium on the Relation between Combinatorics and Other Branches of Mathematics,* 1978, 365-378.

[K3] T.P. Kirkman, "On a problem in combinations", *Cambridge and Dublin Math. J.*, *2*, 1847, 191-204.

[K4] T.P. Kirkman, query, *Lady's and Gentleman's Diary*, *48*, (1950).

[K5] W. Kocay, D.R. Stinson and S.A. Vanstone, "On Strong Starters in Cyclic Groups", preprint.

[K6] D. Konig, *Theorie der Endlichen und Unendlichen Graphen*, Teubner, Leipzig, 1936.

[K7] E.S. Kramer, "Indecomposable Triple Systems", *Discrete Math.*, *8*, (1974) 173-180.

[K8] E.S. Kramer, D. Leavitt and S.S. Magliveras, "Construction Procedures for t-designs and the Existence of New Simple 6-designs", elsewhere in this volume.

[L1] P. Lambossy, "Sur une Maniere de Differencier les Fonctions Cycliques d'une Forme Donnee", *Comment. Math. Helv.*, *3*, (1931) 69-102.

[L2] E. Lamken and S.A. Vanstone, "The Existence of $KS_2(v;u,\lambda)$: Main Constructions", *Utilitas Math.*, to appear.

[L3] E. Lamken and S.A. Vanstone, "The Existence of $KS_2(v;u,\lambda)$: Special Constructions", *Utilitas Math.*, to appear.

[L4] D. Leven and Z. Galil, "NP-Completeness of Finding the Chromatic Index of Regular graphs", *J. Algorithms*, *4*, (1983) 35-44.

[L5] H.R. Lewis and C.H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, 1981.

[L6] R.J. Lipton, "The Beacon Set Approach to Graph Isomorphism", unpublished manuscript, 1979.

[L7] C.C. Lindner, private communications, 1979.

[L8] C.C. Lindner, "A Survey of Embedding Theorems for Steiner Systems", *Annals of Discrete Math.*, *7*, (1980) 175-202.

[L9] C.C. Lindner and T. Evans, *Finite Embedding Theorems for Partial Designs and Algebras*, SMS, University of Montreal, 1977.

[L10]
C.C. Lindner and K.T. Phelps, "A Note on Partial Parallel Classes in Steiner Systems", *Discrete Math.*, *24*, (1978) 109-112.

[L11]
C.C. Lindner and A. Rosa, "Finite Embedding Theorems for Partial Triple Systems with $\lambda>1$", *Ars Combinatoria*, *1*, (1976) 159-166.

[L12]
C.C. Lindner and D.R. Stinson, "The Spectra for the Conjugate Invariant Subgroups of Perpendicular Arrays", *Ars Combinatoria, 18,* (1984) 51-60.

[L13]
G. Lo Faro, "On the Size of Partial Parallel Classes in STS(19) and STS(27)", *Discrete Math., 45 (1983)* 165-179.

[L14]
J.Q. Longyear, "A Survey of Nested Designs", *J. of Statistical Planning and Inference, 5,* (1981) 181-187.

[L15]
J.Q. Longyear, "Nine Small Nested Triple Systems", preprint.

[L16]
J.Q. Longyear, "Nested Group Divisible Designs and Small Nested Designs", preprint.

[L17]
L. Lovász, "Coverings and Colorings of Hypergraphs", *Proceedings of the Fourth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1973, 3-12.

[L18]
E. Luks, "Isomorphism of Graphs of Bounded Valence can be Tested in Polynomial Time", *J. Comp. Sys. Sci., 25,* (1982) 42-65.

[M1] E. Marzewski, "Sur Deux Proprietes des Classes d'Ensembles", *Fund. Math., 33,* (1945) 303-307.

[M2] R.A. Mathon, "Sample Graphs for Isomorphism Testing", *Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1978, 499-517.

[M3] R.A. Mathon, "A Note on the Graph Isomorphism Counting Problem", *Information Processing Letters, 8,* (1979) 131-132.

[M4] R.A. Mathon, private communications, 1984.

[M5] R.A. Mathon, K.T. Phelps and A. Rosa, "Small Steiner Triple Systems and Their Properties", *Ars Combinatoria, 15,* (1983) 3-110; also Mathematics Report 82-OCT-111, Dept. of Mathematical Sciences, McMaster University, 1982.

[M6] R.A. Mathon and A. Rosa, "A Census of Mendelsohn Triple Systems of Order Nine", *Ars Combinatoria, 4,* (1977) 309-315.

[M7] R.A. Mathon and A. Rosa, "A Census of 1-Factorizations of K^3_9: Solutions with Group of Order > 4", *Ars Combinatoria* 16, (1983) 129-148.

[M8] R.A. Mathon and S.A. Vanstone, "On the Existence of Doubly Resolvable Kirkman Systems and Equidistant Permutation Arrays", *Discrete Math.*, *30*, (1980) 157-172.

[M9] E. Mendelsohn, "Perpendicular Arrays of Triple Systems", *Annals of Discrete Math.*, *7*, (1980) 89-94.

[M10] E. Mendelsohn and A. Rosa, "One-factorizations of the Complete Graph: A Survey", *J. Graph Theory*, to appear.

[M11] N.S. Mendelsohn, "A Natural Generalization of Steiner Triple Systems", in: A. Atkin and B. Birch (eds), *Computers in Number Theory*, Academic Press, 1971, 323-338.

[M12] G.L. Miller, "On the $n^{\log n}$ Isomorphism Technique", *Proceedings of the Tenth ACM Symposium on the Theory of Computing*, 1978, 51-58.

[M13] E.H. Moore, "Concerning Triple Systems", *Math. Ann.*, *43*, (1893) 271-285.

[M14] E.H. Moore, "Tactical Memoranda I-III", *American J. Math.*, *18*, (1896) 264-303.

[M15] E.J. Morgan, "Some Small Quasi-Multiple Designs", *Ars Combinatoria*, *3*, (1977) 233-250.

[M16] R.C. Mullin, P.J. Schellenberg, G.H.J. van Rees and S.A. Vanstone, "On the Construction of Perpendicular Arrays", *Utilitas Mathematica*, *18*, (1980) 141-160.

[M17] R.C. Mullin, P.J. Schellenberg, S.A. Vanstone and W.D. Wallis, "On the Existence of Frames", *Discrete Math.*, *37*, (1981) 79-104.

[M18] R.C. Mullin and S.A. Vanstone, "Steiner Systems and Room Squares", *Annals of Discrete Math.*, *7*, (1980) 95-104.

[M19] R.C. Mullin and W.D. Wallis, "The Existence of Room Squares", *Aequationes Math.*, *13*, (1975) 1-7.

[N1] J. Nešetřil, K.T. Phelps and V. Rödl, "On the Achromatic Number of Simple Hypergraphs", *Ars Combinatoria, 16,* (1983) 95-102.

[N2] J. Novák, "A Note on Disjoint Cyclic Triple Systems", in: *Recent Advances in Graph Theory,* Academia, 1975, 439-440.

[O1] W.E. Opencomb, "On the Intricacy of Combinatorial Construction Problems", *Discrete Mathematics, 50,* (1984) 71-97.

[P1] E.S. Page and L.B. Wilson, *An Introduction to Computational Combinatorics,* Cambridge University Press, Cambridge, 1979.

[P2] J. Petersen, "Die Theorie der Regulären Graphen", *Acta Math., 15,* (1891) 193-220.

[P3] L.P. Petrenyuk and A.Y. Petrenyuk, "An Enumeration Method for Non-isomorphic Combinatorial Designs", *Annals of Discrete Math., 7,* (1980) 265-276.

[P4] K.T. Phelps, "A Note on Partial $S(s-1,s+1,v)$ Subsystems of a $S(s,s+1,v)$", *Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory and Computing,* 1978, 581-586.

[P5] K.T. Phelps, "Parallelism in Designs and Strongly Regular Graphs", *Proceedings of the Ninth Manitoba Conference on Numerical Mathematics and Computing,* 1979, 365-384.

[P6] K.T. Phelps, "A Survey of Derived Triple Systems", *Annals of Discrete Math., 7,* (1980) 105-114.

[P7] K.T. Phelps, "On Cyclic Steiner Systems S(3,4,20)", *Annals of Discrete Math., 7,* (1980) 277-300.

[P8] K.T. Phelps, "Partial Parallel Classes and Chromatic Index", unpublished manuscript, 1984.

[P9] K.T. Phelps, private communications, 1984.

[P10] K.T. Phelps and V. Rödl, "On the Algorithmic Complexity of Coloring Simple Hypergraphs and Steiner Triple Systems", *Combinatorica 4 (1),* (1984) 79-88.

[P11] K.T. Phelps and A. Rosa, "2-Chromatic Steiner Quadruple Systems", *European J. Combinatorics, 1,* (1980) 253-258.

[P12] S. Poljak, V. Rödl and D. Turzik, "Complexity of Representation of Graphs by Set Systems", *Discrete Applied Math., 3,* (1981) 301-312.

[R1] D. Raghavarao, *Constructions and Combinatorial Problems in Design of Experiments*, Wiley Publishing Co., New York, 1971.

[R2] D.K. Ray-Chaudhuri and R.M. Wilson, "Solution of Kirkman's Schoolgirl Problem", *Proceeding of Symposium in Pure Math. (American Math. Society), 19,* (1971) 187-203.

[R3] R.C. Read, "Every One a Winner", *Annals of Discrete Math., 2,* (1978) 107-120.

[R4] R.C. Read and D.G. Corneil, "The Graph Isomorphism Disease", *J. Graph Theory, 1,* (1977) 339-363.

[R5] M. Reiss, "Uber eine Steinersche combinatorische Aufgabe, welche im 45sten Bande dieses Journals, Seite 181, gestellt worden ist," *J. reine angew Math., 56,* (1859) 326-344.

[R6] A. Rosa, "On the Chromatic Number of Steiner Triple Systems", *Combinatorial Structures and Their Applications, (Proceedings of the Calgary Conference, 1969),* Gordon and Breach, 1970, 369-371.

[R7] A. Rosa, "Steiner Triple Systems and Their Chromatic Number", *Acta. Fac. Rerun. Natur. Univ. Comcn. Math., 24,* (1970) 159-174.

[R8] A. Rosa, "On Generalized Room Squares", *Proceedings of Problemes Combinatoires et Theorie des Graphes,* Orsay, 1976, 353-358.

[R9] A. Rosa, "Generalized Room Squares and Multidimensional Room Designs", *Proceedings of Second Caribbean Conference on Combinatorics and Computing,* Barbados, 1977, 209-214.

[R10]
A. Rosa, "Intersection Properties of Steiner Systems", *Annals of Discrete Math., 7,* (1980) 115-128.

[S1] J.A.M. Schreuder, "Constructing Timetables for Sport Competitions", *Mathematical Programming Study, 13,* (1980) 58-67.

[S2] D.P. Shaver, *Construction of* (v,k,λ)*-configurations using a Non-enumerative Search Technique,* Ph.D. Thesis, Syracuse University, 1973.

[S3] J.J. Stiffler and J.F. Blake, "An Infinite Class of Generalized Room Squares", *Discrete Math., 12,* (1975) 159-163.

[S4] D.R. Stinson, "A Comparison of Two Invariants for Steiner Triple Systems: Fragments and Trains", *Ars Combinatoria, 16,* (1983) 69-76.

[S5] D.R. Stinson, "Hill-climbing Algorithms for the Construction of Combinatorial Designs", elsewhere in this volume.

[S6] D.R. Stinson, "Isomorphism Testing of Steiner Triple Systems: Canonical Forms", preprint.

[S7] D.R. Stinson, "The Spectrum of Nested Steiner Triple Systems", preprint.

[S8] D.R. Stinson and S.A. Vanstone, "Some Non-isomorphic Kirkman Triple Systems of Order 39 and 51", preprint.

[S9] A.P. Street, "A Survey of Irreducible Balanced Incomplete Block Designs", *Ars Combinatoria,* to appear.

[S10] A.P. Street and W.D. Wallis, *Combinatorial Theory: An Introduction,* The Charles Babbage Research Centre, 1977.

[T1] L. Teirlinck, "On Making Two Steiner Triple Systems Disjoint", *J. Combinatorial Theory (A), 23,* (1977) 349-350.

[T2] M. Tompa, *Hill-climbing, a Feasible Search Technique for the Construction of Combinatorial Configurations,* M.Sc. Thesis, Dept. of Computer Science, University of Toronto, 1975.

[T3] C. Treash, "The Completion of Finite Incomplete Steiner Triple Systems with Applications to Loop Theory", *J. Combinatorial Theory (A), 10,* (1971) 259-265.

[T4] A. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society Series 2, 42,* (1936) 230-265, and 43, 544-546.

[V1] L.G. Valiant, "The Complexity of Computing the Permanent", *Theoretical Computer Science, 8,* (1979) 189-201.

[V2] S.A. Vanstone, "Doubly Resolvable Designs", *Discrete Math., 29,* (1980) 77-86.

[V3] V.G. Vizing, "On an Estimate of the Chromatic Class of a p-graph", (Russian), *Diskret. Analiz., 3,* (1964) 25-30.

[W1] H.S. White, "Triple-Systems as Transformations and Their Paths Among Triads", *Transactions of the American Math. Society, 14,* (1913) 6-13.

[W2] R. M. Wilson, "An Existence Theory for Pairwise Balanced Designs, I: Compositions Theorems and Morphisms", *J. Combinatorial Theory (A), 13,* (1972) 220-245.

[W3] R. M. Wilson, "An Existence Theory for Pairwise Balanced Designs, II: The Structure of PBD-Closed Sets and the Existence Conjectures", *J. Combinatorial Theory (A), 13,* (1972) 246-273.

[W4]R. M. Wilson, "An Existence Theory for Pairwise Balanced Designs, III: Proof of the Existence Conjecture", *J. Combinatorial Theory (A), 18,* (1975) 71-79.

[W5]D.E. Woolbright, "On the Size of Partial Parallel Classes in Steiner Systems", *Annals of Discrete Math., 7,* (1980) 203-211.

Annals of Discrete Mathematics 26 (1985) 137–142
Elsevier Science Publishers B.V. (North-Holland)

Algorithms to Find Directed Packings

Jeremy E. Dawson

Australian National University
Canberra, ACT
AUSTRALIA

Abstract

We discuss computational methods to find (t,v,v) directed packings, i.e. sets of permutations of v symbols such that no t symbols appear in the same order in more than one of the permutations. In so doing, we give algorithms for finding all n-cliques in a given n-partite graph, and in an arbitrary graph. We also discuss an algebraic method for finding directed packings.

1. Introduction

Directed packings are combinatorial structures which are used in the design of statistical experiments and large computer networks [4]. A (t,k,v) directed packing is a collection of ordered k-subsets, called blocks, of a set of cardinality v having the property that no ordered t-tuple occurs in more than one block. An ordered t-tuple is contained in a block if its symbols appear, left to right, in the block. The maximum number of blocks in a (t,k,v) directed packing is denoted $DD(t,k,v)$. If every ordered t-tuple appears once, then we have a 'directed t-design', which is analogous to an ordinary t-design (for $t=2$, see, for example, [5], [6]). However, we look at the case $k=v$, where the analogous t-design is a single complete block. Skillicorn has conjectured that $DD(v-1,v,v) = (v-1)!$ for all v. This has been shown for $v=6$ by computer. A programme, described in Section 2, was written to find all ways of extending a $(t,v-1,v-1)$ directed packing to a (t,v,v) directed packing, by inserting the new symbol somewhere in each block. This programme was applied to the 120 permutations of $\{1,2,3,4,5\}$. In Section 3 we describe an algebraic method of constructing directed packings, which was used in [1] to give results such as $DD(5,7,7)\leq 63$, $DD(5,8,8)\leq 48$ and $DD(5,9,9)\leq 27$.

2. The Main Algorithm

Let a $(t,v-1,v-1)$ directed packing be given, using the symbols $\{1,2,...,v-1\}$; we look for a way to insert v somewhere in each block to give a (t,v,v) directed packing. Write $u=v-1$, calling the original blocks *u-blocks* and the new (extended) blocks *v-blocks*. Let the given (t,u,u) directed packing have n blocks. We define an n-partite graph $G = (V,E)$ with vertex set $V = \{(B,p)\colon B$ a u-block, $1 \le p \le v\}$. Each vertex (B,p) represents a potential v-block, got by inserting v into B in position p; we may refer to this v-block simply as (B,p). We let $[(B,p),(C,q)] \in E$ if the v-blocks (B,p) and (C,q) do not have a t-tuple in common (and so may appear together in a directed packing).

We now need to find a complete n-subgraph of the above n-partite graph. The algorithm therefore falls into two parts.

Part 1: find the adjacency matrix of the graph.

For each pair B,C of u-blocks, we determine the positions p and q for which (B,p) and (C,q) have a t-tuple in common. Such a t-tuple must contain v, so we look for common $(t-1)$-tuples in B and C. Write $s=t-1$. For an s-tuple $(e_1,e_2, \ldots ,e_s)$, found in positions $(p_1,p_2, \ldots ,p_s)$ in B and $(q_1,q_2, \ldots ,q_s)$ in C, (B,p) and (C,q) have a common t-tuple for $0<p \le p_1$ and $0<q \le q_1$, or $p_1<p \le p_2$ and $q_1<q \le q_2$, or ..., or $p_s<p \le v$ and $q_s<q \le v$.

Part 2: find the complete subgraphs.

We have an n-partite graph with parts $B_1,B_2,...,B_n$ (B_i denoting both the u-block of the original packing and the vertex of the graph), and vertices within each part $1,2,...,v$. (The extension to the case where the parts are of unequal size is trivial.) Essentially we look at each possible set of choices of vertex (B_i,p_i) from part B_i; describing such a possibility by the "vertex vector" $(p_1,p_2,...,p_n)$, we look at these vertex vectors in lexicographic order. Our earlier approach was as follows. When we find that a certain vertex vector will not do because say $[(B_i,p_i),(B_j,p_j)] \notin E$ $(i<j)$ then we go to the next vertex vector with a new value for p_j; we check whether $[(B_i,p_i),(B_j,p_j)] \in E$ $(i<j)$ in increasing order of j so that when the vertex vector is changed, starting at position k, we need check whether $[(B_i,p_i),(B_j,p_j)] \in E$ $(i<j)$ for $j \ge k$ only.

However a more efficient approach is available. Upon giving p_i a value, determine which values for each p_j $(j>i)$ are consequently no longer possible ("barred"). Further, if setting p_i causes all values for some p_j to be barred then p_i must be changed. Of course, when we reset p_i, we must remove all bars on $p_j (j>i)$ resulting from the former value of p_i. Thus the following algorithm was used. Actions described other than by elements of PASCAL code or by using subscripts are quoted; {...} denotes a comment. The complete PASCAL programme is available.

```
procedure cvv {change vertex vector at place i} ;
procedure reset(p) {reset bars according to new setting p_i = p} ;
begin for j := i+1 to n do begin
   possible = false ;
   for q := 1 to v do if bar[j,q]=0 or bar[j,q]≥i then begin
    if adj[B_i,p,B_j,q] then begin bar[j,q] := 0 ; possible := true end
                     else bar[j,q] := i ;
   end ;
   if not possible {no value for p_j, so must change p_i} then goto 3
  end
end {reset} ;
begin
  "let S = {p: p_i<p≤v, bar[i,p]=0}"
  if "S=Ø" then i := i-1 ;
   else begin p_i := "min(S)" ;
    if i=n then "output vertex vector"
     else begin reset(p_i) ; i := i+1 ; p_i := 0 end
   end ;
3 : end {cvv} ;
begin for i := 1 to n do for p := 1 to v do bar[i,p] := 0 ;
 i := 1 ; p_i := 0 ;
 repeat cvv until i = 0
end.
```

A further refinement, useful if the graph has many edges, is to have all$[B_i,B_j]$=true if the induced subgraph with vertex set $B_i \cup B_j$ is complete bipartite, and replace the second line of procedure reset by

```
begin for j := i+1 to n do if not all[B_i,B_j] then begin
```

The idea of this refinement was incorporated into a method we used to save storage space. In the problem of packings, if we regard u-blocks B and C as permutations of $\{1,2,...,u\}$, then adj$[B,i,C,j]$ = adj$[I,i,CB^{-1},j]$ (I being the identity permutation), and so we need store only a 3-dimensional array, plus a table of quotients of permutations. We put a zero in this table, quot$[B,C]$=0, where B and C have no common s-tuple and so no (B,p) and (C,q) could have a common t-tuple; otherwise quot$[B,C]=CB^{-1}$.

The programme was run, with the 120 permutations of {1,2,3,4,5}, to examine the conjecture that $DD(5,6,6) = 5!$ It was stopped after about 10000 solutions had been found, when about 1/20th of all possible vertex vectors had been scanned. Probably, therefore, there is a large number of non-isomorphic solutions. By arguments similar to those in [1], the number of occurrences of a given symbol in positions (1,2,...,6) may be (18,30,0,40,10,22), (19,25,10,30,15,21), (20,20,20,20,20,20), (21,15,30,10,25,19) or (22,10,40,0,30,18). Solutions were found with the numbers of occurrences of the various symbols in position 1 being {20,20,20,20,20,20}, {18,20,20,20,20,22}, {18,19,20,20,21,22}, {18,19,19,21,21,22}, {18,19,20,21,21,21} and {19,19,19,20,21,22}.

3. An Algebraic Method

Some other packings were found using an algebraic method. Let the v symbols be the elements of a group G, here written additively (though not necessarily abelian). We look for packings which, whenever they contain a block $B = \{b_1, b_2, \ldots, b_v\}$, also contain $B+g = \{b_1+g, b_2+g, \ldots, b_v+g\}$. To ensure that a system with this property is a (t,v,v) directed packing, it is enough to check that no t-tuple starting with a given symbol is repeated. (In practice, we considered the "initial" block B, and derived any such t-tuples in $B+g$ directly from B.) A similar method is used by Mills in finding BIBDs [3]. We look for a packing which is a disjoint union of as many $\{B+g : g \in G\}$ as possible. The following procedure was used.

1. For each permutation $B = \{b_1, b_2, \ldots, b_v\}$, construct the list $\{(e_2-e_1, e_3-e_1, \ldots, e_t-e_1) : (e_1, e_2, \ldots, e_t) \text{ is a } t\text{-tuple of } B\}$ (since $(0, e_2-e_1, e_3-e_1, \ldots, e_t-e_1)$ is a t-tuple in $B-e_1$).

2. For each pair B,C of permutations of G compare their lists, setting adj$[B,C]$:= true if they do not intersect.

We now have a graph with vertex set the permutations of G, and edge set $\{[B,C] : \{B+g : g \in G\} \cup \{C+g : g \in G\}$ is a directed packing$\}$ of which we want the largest complete subgraph. A recursive procedure was used for this, starting with a list of (vertices of) complete subgraphs of order 2.

3. For $n = 2,3,\ldots$, do the following, which gives a list of $n+1$- cliques (complete subgraphs of order $n+1$) from the list of the n-cliques. Take the list of n-cliques, in which the vertices of each n-clique are listed in ascending order, and the n-cliques are listed in increasing lexicographic order. Consider this list in segments, each segment being the set of graphs differing only in the last

vertex.

for the segment $\{\{v_1,v_2,...,v_{n-i},w_i\} : i=1,2,...,m\}$ do
  for i := 1 to $m-1$ do for j := $i+1$ to m do
    if adj$[w_i,w_j]$ then write $(v_1,v_2,...,v_{n-1},w_i,w_j)$ {onto the list of $n+1$-cliques}.

We now have a list of $n+1$-cliques in lexicographic order.

This approach was taken further by considering a group A of automorphisms of G. An initial block $B = \{b_1,b_2,...,b_v\}$ gives rise to blocks $\theta(B+g) = \{\theta(b_1+g),\theta(b_2+g),...,\theta(b_v+g)\}$ for each $g \in G$ and $\theta \in A$. If the stabilizer in A of 0 has p orbits on $G \backslash \{0\}$, then we need only ensure that no t-tuple whose first symbol is 0 and whose next different symbol is one of a given set of representatives of these orbits is repeated. For example, where G and A are the additive and multiplicative groups of a field, then we need only check t-tuples whose first symbol is 0 and whose first non-zero symbol is 1; furthermore, in this case each t-tuple of B gives rise to exactly one such t-tuple in some block $\theta(B+g)$. This method has been used to find lower bounds for some values of $DD(4,v,v)$ and $DD(5,v,v)$; the actual G and A and the results are reported in [1]. More generally, we could consider any group of permutations acting on the set of symbols, and use the transitivity structure of G to simplify the task of ensuring that no t-tuple is repeated. Such a technique is used by Mills to find block designs [2].

REFERENCES

[1] J.E. Dawson, J.R. Seberry and D.B. Skillicorn, "The directed packing numbers $DD(t,v,v)$", *Combinatorica,* to appear.

[2] W.H. Mills, "A new 5-design", *Ars Combinatoria* 6 (1978) 193-195.

[3] W.H. Mills, "The construction of balanced incomplete block designs", Proceedings of the 10th Southeastern Conference on Combinatorics, Graph Theory and Computing, *Congressus Numerantium* 23 (1979) 73-86.

[4] D.B. Skillicorn, "Directed packings and coverings with computer applications", Ph.D. thesis, University of Manitoba, 1981.

[5] D.B. Skillicorn, "Directed coverings and packings of pairs and quadruples", Combinatorial Mathematics IX, Lecture Notes in Mathematics 952 (Springer-Verlag 1982) 387-391.

[6] D.J. Street and J.R. Seberry, "All DBIBDs with block size four exist", *Utilitas Mathematica* 18 (1980) 27-34.

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 143–150

Four Orthogonal One-Factorizations on Ten Points

J.H. Dinitz

Department of Mathematics
University of Vermont
Burlington, Vermont
U.S.A.

W.D. Wallis

Department of Mathematics, Statistics
and Computer Science
University of Newcastle
Newcastle, N.S.W.
AUSTRALIA

Abstract

Using list processing techniques, an exhaustive search was made for orthogonal one-factorizations of K_{10}. As a result we have found that, up to isomorphism, there is exactly one set of four mutually orthogonal one-factorizations of K_{10}, and exactly 267 sets of three mutually orthogonal one-factorizations of K_{10}.

1. Introduction

Let G be a graph with an even number of vertices. A *one-factor* in G is a set of (pairwise disjoint) edges which between them contain each vertex exactly once. A *one-factorization* is a way of decomposing the edges of G into pairwise disjoint one-factors. In particular it is well-known that the complete graph K_{2n} on $2n$ vertices has a one-factorization, which consists of $2n$-1 factors. If $F = \{F_1, F_2, ..., F_k\}$ and $G = \{G_1, G_2, ..., G_k\}$ are one-factorizations of the same graph G, we say that F and G are *isomorphic* if there exists a map ϕ which permutes the vertices of G and a map ψ which permutes the integers $\{1, 2, ..., k\}$ such that for all i, $F_i\phi = G_{i\psi}$ ($F_i\phi$ is the graph derived from F_i by replacing each edge (x, y) by $(x\phi, y\phi)$). We usually refer to ϕ as "the isomorphism", the existence of a suitable map ψ being implicit.

Two one-factorizations $\{F_1, F_2, ..., F_k\}$ and $\{G_1, G_2, ..., G_k\}$ of G are called *orthogonal* if, for every i and j, F_i and G_j have at most one edge in common. Orthogonal one-factorizations of complete graphs correspond to Room Squares (see [6, 7]), orthogonal one-factorizations of regular complete bipartite graphs correspond to Latin squares (see [8]). In this paper we discuss orthogonal one-factorizations of K_4, K_6, K_8 and especially of K_{10}, where we have run an exhaustive search for sets of orthogonal one-factorizations.

If F, G, F' and G' are factorizations of the same graph, F is orthogonal to G, and there is an isomorphism which takes F to F' and simultaneously takes G to G', we can consider the pairs $\{F, G\}$ and $\{F', G'\}$ to be the same up to isomorphism. If G is a complete graph, this means that the corresponding Room squares would be isomorphic also.

Let $\nu(r)$ be the maximum possible number of mutually orthogonal one-factorizations of K_r, r even. It is easy to see that $\nu(r) \leq r - 3$, but no better upper bound has been found in general. However, no case is known where $\nu(r)$ is greater than $\frac{r}{2} - 1$, and some authors believe this is an upper bound for all r. It is known that $\nu(r) \geq \frac{r}{2} - 1$ when $r - 1$ is a prime power congruent to 3(modulo 4). However $\nu(6) = 1$ (see the next section), so $\nu(r) = \frac{r}{2} - 1$ cannot always be achieved, even when $r - 1$ is a prime. The main result of this paper, that $\nu(10) = \frac{10}{2} - 1 = 4$, does however lend support to the conjecture that $\nu(r) \leq \frac{r}{2} - 1$ and that this bound can almost always be attained.

For further information on $\nu(r)$, see [2, 3, 6, 7]. In particular, it is shown in [6] that $\nu(r)$ approaches infinity with r.

Our aim here is to study orthogonal factorizations of K_{10} fully. Not only do we wish to evaluate $\nu(10)$, but we hope that a full study will aid understanding of the behaviour of orthogonal one-factorizations in general.

2. Small Orders

The one-factorizations of small complete graphs are easily studied. For K_2 and K_4 there is only one factorization. K_6 admits fifteen one-factors and six one-factorizations; each factor lies in exactly two factorizations and any two factorizations have exactly one factor in common; the six factorizations are isomorphic. So there is one factorization up to isomorphism, and there are no pairs of orthogonal factorizations, up to order 6: $\nu(2) = \nu(4) = \nu(6) = 1$.

For K_8 the situation is more interesting. A complete analysis is given in [9]. There are six non-isomorphic one-factorizations, which we list in Table 1. We shall call them F_1, F_2, F_3, F_4, F_5 and F_6. Up to isomorphism there are four factorizations orthogonal to F_1, three to F_4, two to F_5 and one to F_6; those orthogonal to F_1 are isomorphic to F_1, F_4, F_5 and F_6 respectively. Those orthogonal to F_4 are isomorphic to F_1, F_4 and F_5 respectively; those orthogonal to F_5 are isomorphic to F_1 and F_4 respectively; the one orthogonal to F_6 is isomorphic to F_1. Allowing for double counting (since {F, G} and {G, F} are the same pair) we have six pairs up to isomorphism. Interestingly, there are no cases of non-isomorphic pairs {F, G} and {F, H} where G is isomorphic to H, but such pairs appear for higher orders. There is precisely one set of three mutually orthogonal factorizations up to isomorphism (isomorphic to F_1, F_1 and F_6), and no set of four. So $\nu(8) = 3$.

All One-Factorizations of K_8
Table 1

01	23	45	67	01	23	45	67
02	13	46	57	02	13	46	57
03	12	47	56	03	12	47	56
04	15	26	37	04	15	26	37
05	14	27	36	05	14	27	36
06	17	24	35	06	17	25	34
07	16	25	34	07	16	24	35
	F_1				F_2		
01	23	45	67	01	23	45	67
02	13	46	57	02	13	46	57
03	12	47	56	03	12	47	56
04	16	25	37	04	16	27	35
05	17	26	34	05	17	26	34
06	14	27	35	06	14	25	37
07	15	24	36	07	15	24	36
	F_3				F_4		
01	23	45	67	01	23	45	67
02	13	46	57	02	14	36	57
03	14	27	56	03	16	25	47
04	16	25	37	04	17	26	35
05	17	26	34	05	12	37	46
06	12	35	47	06	15	27	34
07	15	24	36	07	13	24	56
	F_5				F_6		

3. Order Ten

An exhaustive search for orthogonal one-factorizations of K_{10} was made, with the main result being that $\nu(10) = 4$. In this section we shall discuss the method employed in this search and some of the findings.

In his thesis [4], Gelling determined the complete set of non-isomorphic one-factorizations of K_{10} (see also [5]). These are 396 in all, which we shall denote as $G_1, G_2, \ldots, G_{396}$ (in Gelling's order). Our search begins by choosing a one-factorization, G_n say. We find all one-factorizations orthogonal to G_n, and then check this list for mutual orthogonality. If any set of orthogonal one-factorizations contains a factorization isomorphic to G_n, then applying the inverse isomorphism to all the factorizations will produce an isomorphic set which contains G_n itself; so if we let n range from 1 to 396 we shall obtain a complete list of all isomorphism classes of orthogonal one-factorizations of K_{10}. (The list could contain some repetitions, as no isomorph-rejection has been carried out after the selection of G_n; but the number of repetitions should be very small, since the one-factorization of K_{10} mostly have small automorphism groups - 298 of them have the identity group [4]).

We used a Fortran program which employed three subroutines: WINNOW, RS9S and ORTHOG. Let $G_n = \{g_1, g_2, \ldots, g_9\}$ be the nth one-factorization on Gelling's list. The subroutine WINNOW reads in all 945 one-factors of K_{10} and outputs those which could possibly be contained in a one-factorization orthogonal to G_n. That is, if W is a one-factor of K_{10}, W will be output if and only if W and g_i have at most one edge in common for $i = 1, 2, \ldots, 9$.

The subroutine RS9S reads in the one-factors supplied by WINNOW. From this list it constructs all possible one-factorizations using only these one-factors. So it constructs the one-factorizations orthogonal to G_n. At this point some duplication could occur - RS9S might produce two factorizations, K and L say, such that some isomorphism χ exists which maps G_n to itself and also maps K to L. As explained above, the number of such occurrences should be small, and it is much cheaper (in terms of CPU time) to allow such duplications to occur than to conduct isomorph-rejection at this stage.

Finally, the subroutine ORTHOG checks pairs of one-factorizations from RS9S for orthogonality. Then if two one-factorizations K and L are found to be orthogonal to each other, G_n and K and L form a set of three mutually orthogonal one-factorizations, and they are output.

The number of triples is sufficiently small for further work to be done most efficiently by hand. We did this and found our main theorem.

Theorem 1: There is exactly one set of four mutually orthogonal one-factorizations of K_{10}, up to isomorphism. This set does not extend to a set of five mutually orthogonal one-factorizations.

The set of four factorizations is shown in Table 2.

Four Orthogonal One-Factorizations of K_{10}
Table 2.

01	23	45	67	89	01	29	36	48	57
02	13	46	58	79	02	15	34	69	78
03	12	47	59	68	03	16	28	45	79
04	16	25	39	78	04	17	26	35	89
05	18	24	37	69	05	14	27	39	68
06	19	27	35	48	06	12	37	49	58
07	15	28	36	49	07	19	25	38	46
08	17	29	34	56	08	13	24	59	67
09	14	26	38	57	09	18	23	47	56
		F_1					F_2		
01	26	39	47	58	01	25	34	68	79
02	14	37	56	89	02	18	35	49	67
03	17	25	48	69	03	15	27	46	89
04	18	27	36	59	04	13	28	57	69
05	19	28	34	67	05	16	29	38	47
06	15	24	38	79	06	14	23	59	78
07	13	29	45	68	07	12	39	48	56
08	16	23	49	67	08	19	26	37	45
09	12	35	46	78	09	17	24	36	58
		F_3					F_4		

The uniqueness may be checked by computer (in about 32 hours CPU time).

The four factorizations have an interesting structure. F_1 is isomorphic to G_{380} in Gelling's list, while F_2, F_3 and F_4 are all isomorphic to G_{377}. The set has automorphism group of order 3, generated by $\sigma = (013)(476)(598)$ which is an automorphism of F_1 and swaps $F_4 \to F_3 \to F_2 \to F_4$.

We note again that our result says that $\nu(10) = 4$. This is significant in that it is the first known example of a number r with $r \equiv 2$ (modulo 4) and $\nu(r) \geq \frac{r}{2} - 1$.

By use of invariants of one-factorizations it is possible to compute the exact number of non-isomorphic sets of three mutually orthogonal one-factorizations of K_{10}. This number is 267. A listing of these triples and a description of the method will appear in a later paper. Beaman [1] determined that there are exactly 511,562 distinct ordered pairs of orthogonal 1-factorizations (non-isomorphic Room squares) and exactly 257,630 unordered pairs (inequivalent Room squares).

Notice that our computational approach was essentially a list-processing one. Backtrack methods were tried experimentally, but are slower by a considerable margin (by a factor of over 100 in the WINNOW process).

Acknowledgement

The authors wish to thank the Academic Computing Center at the University of Vermont for computing support and assistance.

Addendum

For a complete list of all 267 sets of three mutually orthogonal one factorizations of K_{10}, see

D.S. Archdeacon, J.H. Dinitz, and W.D. Wallis, "Sets of pairwise orthogonal 1-factorizations of K_{10}", *Congressus Numerantium,* to appear.

References

[1] I.R. Beaman, "Variability of Room squares", Ph.D. Thesis, University of Newcastle (1977).

[2] J.H. Dinitz, "Pairwise orthogonal symmetric Latin squares", *Congressus Numerantium* 32 (1981), 261-265.

[3] J.H. Dinitz, "Room n-cubes of low order", *Journal of the Australian Mathematics Society* A36 (1984) 237-252.

[4] E.N. Gelling, "On 1-factorizations of the complete graph and the relationship to round robin schedules", MA Thesis, University of Victoria (1973).

[5] E.N. Gelling and R.E. Odeh, "On 1-factorizations of the complete graph and the relationship to round robin schedules", *Proceedings of the Third Manitoba Conference on Numerical Mathematics and Computing,* Winnipeg, (1973), 213-221.

[6] K.B. Gross, R.C. Mullin and W.D. Wallis, "On the number of pairwise orthogonal symmetric Latin squares", *Utilitas Mathematica* 4 (1973), 239-251.

[7] J.D. Horton, "Room squares and one-factorizations", *Aequationes Mathematica* 22 (1981), 56-63.

[8] S.H.Y. Hung and N.S. Mendelsohn, "On Howell designs", *Journal of Combinatorial Theory* 16A (1974), 174-198.

[9] W.D. Wallis, A.P. Street and J.S. Wallis, *Combinatorics: Room Squares, Sum-free Sets, Hadamard Matrices*. Springer-Verlag, Berlin (1972).

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 151–164

A Problem of Lines and Intersections With an Application to Switching Networks

D. Z. Du

University of California
Santa Barbara, California

F. K. Hwang

AT & T Bell Laboratories
Murray Hill, New Jersey 07974
U.S.A.

G.W. Richards

AT & T Bell Laboratories
Naperville, Illinois 60566
U.S.A.

Abstract

We consider lines in space and their intersections. The lines are partitioned into classes where lines in the same class are parallel. A node is an intersection through which a line of each class passes. We look for the minimum number of lines in r classes such that the number of nodes they generate exceeds the number of lines. We mention an application of the problem to the construction of nonblocking switching networks with details to come in a subsequent paper.

1. Introduction

Consider a set of lines in the plane. Lines are in the same class if they are parallel. Define $f(r)$ to be the minimum number of lines, given that the lines are in r classes, which generate at least $f(r)+1$ nodes, where a node is defined as a point with r lines passing through it (one from each class). We give upper bounds and lower bounds for $f(r)$. We then study a similar problem for lines in higher-dimensional space. We show how this problem is related to a construction for nonblocking switching networks.

2. An Upper Bound on $f(r)$

Define $U(r) = \frac{2}{3} r^3 - 2r^2 + \frac{16}{3} r - 3.$

Theorem 1. $f(r) \le U(r)$ for all $r \ge 1$.

Proof: The proof is by a construction of $U(r) + 1$ nodes formed by the intersection of only $U(r)$ lines. We first give a description of the r slopes which define the classes. For r odd the slopes are $\pm 1, \pm 2, \ldots, \pm(r-1)/2$ and ∞ (vertical); for r even the slopes are $\pm 1, \pm 2, \ldots, \pm \frac{r-2}{2}$, ∞ and 0. The set of nodes $S(r)$ will be taken from a square array of points and will be contained in a convex polygon. Furthermore, the set will be symmetric with respect to both the vertical direction and the horizontal direction. Therefore, it suffices to describe the set by giving the length of the rows in the upper half (as the number of rows has the same parity as r for $r \ge 2$, the description includes the middle row for odd $r \ge 3$). Let $H(r)$ denote the upper half of $S(r)$. Let $(x)^k$ denote k consecutive rows of length x. For r even:

$$H(2) = (3)$$

$$H(r) = H(r-2), (2r-5)^{\frac{r}{2}-2}, (2r-3)^{\frac{r}{2}-1}, (2r-1)\,; \; r = 4, 6, 8, \cdots$$

It is easily verified that $S(2)$ has $U(2)+1 = 6$ nodes but only $U(2) = 5$ lines (two horizontal and three vertical). Assume that $S(r-2)$ has $U(r-2)$ lines and $U(r-2)+1$ nodes. We prove that $S(r)$ has $U(r)$ lines and $U(r)+1$ nodes.

The number of additional nodes in $S(r) - S(r-2)$ is

$$2[(\frac{r}{2} - 2)(2r-5) + (\frac{r}{2} - 1)(2r-3) + (2r-1)]$$
$$= 4r^2 - 16r + 24$$
$$= [U(r)+1] - [U(r-2)+1].$$

Next we count the number of additional lines in $S(r) - S(r-2)$. Clearly there are four additional columns and $4(\frac{r}{2} - 1)$ additional rows. We now show that each class of $S(r-2)$, except columns and rows, has $4(\frac{r}{2} - 1)$ additional lines in $S(r)$. We will illustrate our argument by referring to the example of $S(6)$.

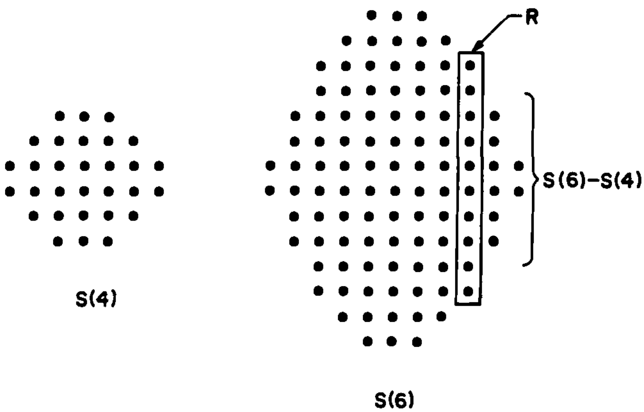


Figure 1. $S(4)$ and $S(6)$

Let C be a class, neither columns nor rows, in $S(r-2)$ with positive slope. We note that all lines of C pass through the upper half except the line which passes through the rightmost point of the lower middle row. Now consider the lines of class C in $S(r)$ which is constructed by inserting $S(r) - S(r-2)$ between the two halves of $S(r-2)$. Let R denote the column in $S(r)$ which starts with the rightmost point of the upper half of $S(r-2)$ and ends with the rightmost point of the lower half of $S(r-2)$. A line of class C either stays to the left of R or intersects with a node on R (our choice of slopes does not allow a line to slip through two nodes on R). If it stays to the left or intersects with the starting point or the ending point of R, then it is a line counted in $S(r-2)$. Therefore the new lines in C are those which intersect with the nodes on R except the starting and the ending one. There are $4(\frac{r}{2} - 1)$ such nodes (one for each new row), hence $4(\frac{r}{2} - 1)$ new lines of C. By symmetry, we can say the same for a class with negative slope.

Finally, we count the number of lines in the two new classes with slopes $\pm \frac{r-2}{2}$. Take the upper center row (of length $2r-1$). Then there is a line of slope $\frac{r}{2} - 1$ passing through every node and $\frac{r}{2} - 2$ lines of slope $\frac{r}{2} - 1$ passing through between every pair of nodes. It is easily seen that the only line of slope $\frac{r}{2} - 1$ not intersecting that row is the one passing through the rightmost node of the lower center row. Thus there is a total of $(2r-2)(\frac{r}{2} - 1) + 2$ lines of slope $\frac{r}{2} - 1$. By symmetry, the number of lines of slope $(-\frac{r}{2} + 1)$ is the same.

Summing up, the total number of new lines is

$$4+(r-3)\,[4(\frac{r}{2}-1)] \;+\; 2[(2r-2)\,(\frac{r}{2}-1)+2]$$
$$= 4r^2-16r+24$$
$$= U(r)-U(r-2).$$

For r odd, $S(1)$, $S(3)$ and $H(5)$ are shown in Fig. 2 ($H(r)$ does not include the middle row):

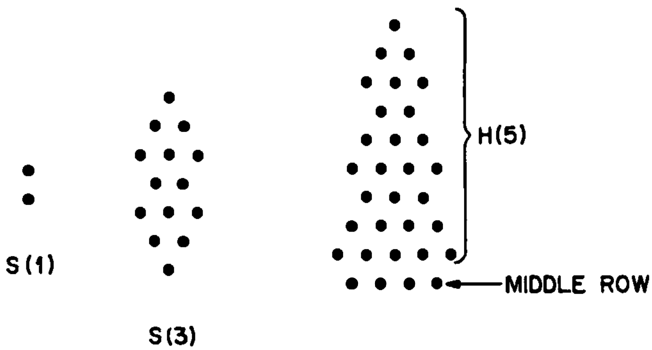


Figure 2. $S(1)$, $S(3)$ and $H(5)$

For $r \geq 3$, $H(r)$ can be constructed recursively by putting together $H(r-2)$, $(r-3,\, r-2)^{(r-3)/2}$, $r-1$, $(r-2,\, r-1)^{(r-3)/2}$, r while the size for the middle row is $r-1$.

The number of additional nodes in $S(r) - S(r-2)$ is

$$2[(r-3+r-2)\,\frac{r-3}{2} + r-1+(r-2+r-1)\,\frac{r-3}{2} + r] + r-1-(r-3)$$
$$= 4r^2-16r+24$$
$$= [U(r)+1]-[U(r-2)+1].$$

The number of additional lines is the same as the even r case and can be obtained by an analogous argument.

3. Lower bounds on $f(r)$

Define $N(l,r)$ to be the maximum number of nodes generated by l lines in r classes. To prove $f(r) \geq l'$ it suffices to prove $N(l,r) < l+1$ for all $l < l'$.

Lemma 1. Suppose that each slope contains at least two lines. Then

$$N(l, r) \leq N(l-2r, r) + 2(l-r)/(r-1).$$

Proof. All the nodes must lie within a convex polygon whose boundaries are the boundary lines of the r slopes. We first count the number of nodes on the boundary. There are at most $2r$ boundary lines forming at most $2r$ intersections between themselves on the boundary (these are the extreme points of the polygon). Any nonboundary line can intersect the boundary at most twice. Hence the total number of intersections is at most

$$2r + 2(l-2r) = 2(l-r).$$

This way of counting intersections counts $r-1$ intersections for each node. Hence the number of nodes on the boundary is at most $2(l-r)/(r-1)$.

If a boundary line of a given slope is not a boundary line of the polygon, then it must lie outside of the polygon and contains no node. Hence throwing away the two boundary lines of each slope can only throw away nodes on the boundary. Lemma 1 is proved.

Corollary. $N(l, r) \leq \dfrac{l^2+r^2-2r}{2r(r-1)}$ for $r \geq 2$.

Proof: Consider a configuration achieving $N(l, r)$ and let l_i denote the number of lines of the ith slope. Suppose $\min l_i = 2m$ for some $m \geq 1$. By Lemma 1

$$\begin{aligned} N(l, r) &\leq \frac{2(l-r)}{r-1} + \frac{2(l-3r)}{r-1} + \cdots + \frac{2[l-(2m-1)r]}{r-1} \\ &= \frac{2(ml-m^2r)}{r-1} \quad \text{(achieving maximum at } m = l/2r) \\ &\leq \frac{l^2}{2r(r-1)} \\ &\leq \frac{l^2+r^2-2r}{2r(r-1)} \quad (r \geq 2) \end{aligned}$$

Suppose $\min l_i = 2m+1$ for some $m \geq 0$. If $m \geq 1$, then by Lemma 1

$$N(l,r) \leq \frac{2(ml-m^2r)}{r-1} + \frac{l-2mr-1}{r-1}$$

where the last term represents $N(l-2mr, r)$ for $\min l_i = 1$. As it turns out, the above inequality is also valid for $m = 0$. But the right-hand-side achieves its maximum at $m = \frac{l-r}{2r}$. Hence

$$N(l,r) \leq \frac{2[\frac{l-r}{2r}\, l - (\frac{l-r}{2r})^2 r]}{r-1} + \frac{l-2r(\frac{l-r}{2r})-1}{r-1}$$

$$= \frac{l^2+r^2-2r}{2r(r-1)}$$

Theorem 2. $N(l,r) < l+1$ for all $l < r(r-1+\sqrt{r^2-2r+2})$

Proof: It is easily verified that $l < r(r-1+\sqrt{r^2-2r+2})$ implies $\frac{l^2+r^2-2r}{2r(r-1)} < l+1$.

Hence $N(l,r) < l+1$.

Corollary. $f(r) \geq 2r(r-1)+1$.

From the Corollary $f(2) \geq 5$, $f(3) \geq 13$. From Theorem 1, $f(2) \leq u(2) = 5$, $f(3) \leq u(3) = 13$. Hence $f(2) = 5$, $f(3) = 13$.

We next derive an improved bound on $f(r)$ that holds asymptotically. Let L be the line system consisting of l_i lines of the ith slope. Since every two nonparallel lines intersect, there are $\sum_{i,j} l_i l_j$ intersections and each node accounts for $\binom{r}{2}$ intersections. However, not all intersections occur at nodes. We call an intersection not occurring at a node an *off-intersection*. If we can show that the number of off-intersections is at least x, then the number of nodes is at most

$$(\sum_{i,j} l_i l_j - x) / \binom{r}{2}.$$

For each node v there are r lines passing through it. On each such line, v usually has two adjacent nodes unless v is an endpoint; then v may have one or zero adjacent nodes. Define $S(v)$ to be the set of $2r$ line segments incident on v where a line segment is from v to an adjacent node if there is one along a half line, or just the half line if there is none. Note that each line segment is counted at most twice when we scan $S(v)$ over v. A line segment (u, v) is counted exactly twice, once in $S(u)$ and once in $S(v)$.

We will count the number of off-intersections involving segments in $S(v)$ for each choice of v. To do this, we will examine all the line segments in $S(v)$ together, but we only consider their intersections with lines from two classes at a time. Thus consider the $l_i \times l_j$ grid formed by the lines from class i and class j. Notice that all nodes must be grid points from this grid, but not all grid points need be nodes. To avoid certain boundary effects in the counting, we first augment the grid by adding $2y$ lines to each class, with y of them on each side of the original lines in the class, where y is a suitably large constant (independent of i and j) to be specified later. Then any segment in $S(v)$ that has fewer than y off-intersections in the enlarged grid must pass through some grid point. We will only be concerned with the off-intersections on such a segment that occur between v and the first grid point encountered by the segment (which may or may not be a node). More precisely, for $i = 0, 1, \ldots, y-1$, we will bound the number of segments in $S(v)$ that can have exactly i such intersections between v and the first grid point encountered by the segment, by bounding the number of grid points that can be reached from v by a line segment that passes through no other grid point and that intersects exactly i lines of the grid. In particular, we immediately see that there are at most 8 grid points reachable in this way from v with no intermediate intersections. Referring to Figure 3, it is not hard to see that for $i = 1, 2, \ldots, y-1$ there are at most $4(i+1)$ grid points reachable from v by a line segment that encounters no other grid point and that intersects exactly i lines of the grid. Thus, if the number $2r$ of line segments in $S(v)$ satisfies

$$(*) \qquad 2r \geq 8 + \sum_{i=1}^{y-1} 4(i+1)$$

then the number of off-intersections for $S(v)$ in this grid must be at least

$$\sum_{i=1}^{y-1} 4(i+1)i + \left[2r - 8 - \sum_{i=1}^{y-1} 4(i+1)\right]y$$

$$= (2r-8)y - \frac{2y(y-1)(y+4)}{3} = \alpha$$

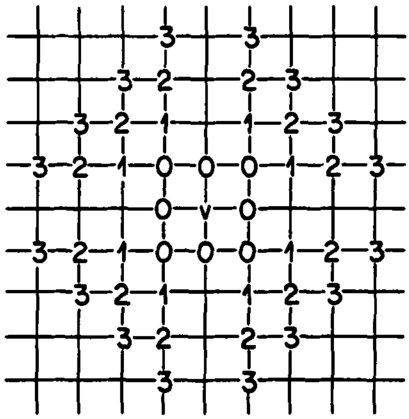


Figure 3

Straightforward algebra shows that for (*) to be satisfied we must have $y^2+y+2 \leq r$ or, equivalently,

$$y \leq \frac{\sqrt{4r-7}-1}{2}$$

Thus we simply take y to be the largest integer satisfying this inequality.

There are $\binom{r}{2}$ ways of choosing two of the r classes, and each off-intersection will be counted at most $r-2$ times for $S(v)$ (once for each grid involving a pair of classes such that one class is the one for the line intersecting the segment from $S(v)$ and the other class is one of the $r-2$ classes not involved in the intersection). Therefore, the line segments in $S(v)$ are involved in at least $\alpha \binom{r}{2} / (r-2)$ off-intersections in the augmented line system L' with the additional $2y$ lines in each class.

Theorem 3. $N(l, r) < l+1$ if $l < \frac{r}{2}(r\beta-4y+\sqrt{(r\beta-4y)^2+4\beta})$ where $\beta = 1+\alpha/2(r-2)$.

Proof: Suppose that L contains x nodes. Then these nodes account for $x\binom{r}{2}$ intersections and $x\binom{r}{2}\alpha/2(r-2)$ off-intersections (one off-intersection can be counted once in $S(v)$ and once in $S(u)$) in L'. Furthermore, the $4\binom{r}{2}y^2$ intersections involving pairs of added lines are not counted in either of the above two terms. Therefore

$$\sum_{i \neq j} (l_i + 2y)(l_j + 2y) \geq x\binom{r}{2} + x\binom{r}{2}\alpha/2(r-2) + 4\binom{r}{2}y^2 \ ,$$

or

$$\sum_{i \neq j} l_i l_j + 2y(r-1)l \geq x\binom{r}{2}\beta$$

It is well known (see p. 52 of [3]) that

$$\left(\frac{\sum_{i \neq j} l_i l_j}{\binom{r}{2}}\right)^{1/2} \leq \frac{l}{r}.$$

Hence

$$x \leq \frac{l^2 + 4yrl}{r^2\beta}\ .$$

Consequently, $N(l, r) < l+1$ if

$$\frac{l^2 + 4yrl}{r^2\beta} < l+1\ ,$$

or if,

$$l < \frac{r}{2}\,(r\beta - 4y + \sqrt{(r\beta - 4y)^2 + 4\beta}\ .$$

Corollary 1. $f(r) \geq r(r\beta - 4y) + 1$.

For r large, $y \to r^{1/2}$, $\alpha \to \frac{4}{3}\, r^{3/2}$ and $\beta \to \frac{2}{3}\, r^{1/2}$. Hence

Corollary 2. $f(r) \to \frac{2}{3}\, r^{5/2}$ for r large.

4. Lines in higher-dimension spaces

In this section we consider lines in a d-dimensional space. Let $F(d)$ denote the minimum number of lines in a d-dimensional space which generate at least $F(d)+1$ nodes, given that a line must be parallel to one of the d axes. Define $U(d) = \sum_{i=1}^{d} i^i$.

Theorem 4. $F(d) \leq U(d)$.

Proof. We give construction of a set $S(d)$ of $U(d)+1$ nodes formed by only $U(d)$ lines. We denote a point in d-dimensional space by d coordinates.

Define $S(1) = \{(1), (2)\}$. Let $S'(d)$ be the projection of $S(d-1)$ into d-dimensional space by adding $d+1$ as the dth coordinate to every node in $S(d-1)$.

Define $C(d) = \{(x_1, x_2, \ldots, x_d);\ 1 \leq x_i \leq d\ ,\ 1 \leq i \leq d\}$.

Define $S(d) = C(d) \cup S'(d)$.

It is easily verified by induction that $S(d)$ is what we want.

We now give a lower bound for $F(d)$. We first prove a lemma.

Define

$$H(k) = \frac{n}{k} + k(d-1)\left(\frac{n}{k}\right)^{1-\frac{1}{d-1}}$$

Lemma 2. $H(k) \geq H(n^{\frac{1}{d}}) = dn^{1-\frac{1}{d}}$.

Proof: Set

$$0 = H'(k) = \frac{-n}{k^2} + (d-1)\left(\frac{n}{k}\right)^{1-\frac{1}{d-1}} + k(d-1)\left(1-\frac{1}{d-1}\right)\left(\frac{n}{k}\right)^{-\frac{1}{d-1}}\left(\frac{-n}{k^2}\right)$$

$$= -\frac{n}{k^2} + \left(\frac{n}{k}\right)^{1-\frac{1}{d-1}}$$

Then clearly, $k^* = n^{\frac{1}{d}}$ minimizes $H(k)$. But

$$H(k^*) = n^{1-\frac{1}{d}} + n^{\frac{1}{d}}(d-1)n^{(1-\frac{1}{d})(1-\frac{1}{d-1})}$$

$$= n^{1-\frac{1}{d}} + (d-1)n^{1-\frac{1}{d}} = dn^{1-\frac{1}{d}}$$

Theorem 5. $F(d) \geq d^d$

Proof: It suffices to prove that any n nodes in a d-dimensional space must involve at least $dn^{1-\frac{1}{d}}$ lines. For if this is true, then the number of nodes can exceed the number of lines only if $n > dn^{1-\frac{1}{d}}$, or equivalently $n > d^d$. This implies $F(d) \geq d^d$.

Partition the n nodes according to the first coordinate into, say k hyperplanes each of which is orthogonal to the first axis. Suppose that the ith hyperplane has n_i nodes. Then there are at least max $\{n_1, \ldots, n_k\}$ lines parallel to the first axis. By induction, these n nodes have at least

$$g(n_1, \ldots, n_k) = \max\{n_1, \ldots, n_k\} + \sum_{i=1}^{k} (d-1)n_i^{1-\frac{1}{d-1}}$$

lines passing through them.

Claim. $g(n_1, \ldots, n_k) \geq dn^{1-\frac{1}{d}}$.

The claim is true for $k = 1$ by Lemma 2. We prove the general case by induction on k. Suppose that $(n_1^*, \ldots, n_k^*)$ minimizes g.

Case (i). $n_1^* = \ldots = n_k^*$. Then the claim follows from Lemma 2.

Case (ii). If n_i^* are not all equal, assume $n_1^* \geq \ldots \geq n_k^*$. Then there must exist an l, $1 \leq l \leq k-1$, such that $n_1^* = \ldots = n_l^* > n_{l+1}^* \geq \ldots \geq n_k^*$. We consider three subcases:

a. $n_k^* = 0$. Then

$$g(n_1^*, \ldots, n_k^*) = g(n_1^*, \ldots, n_{k-1}^*) \geq dn^{1-\frac{1}{d}} \text{ by induction.}$$

b. $n_k^* > 0$, $l \neq k-1$. Then we can reduce g by increasing n_{l+1}^* and decreasing n_k^* since $\sum_{i=1}^{k} (d-1)n_i^{1-\frac{1}{d-1}}$ is easily seen to be a Schur concave function and the new set of n_i^* majorizes the old set (see p. 89 of [3]). But such a reduction contradicts the assumption that $(n_1^*, \ldots, n_k^*)$ is minimum.

c. $l = k-1$, $n_{k-1}^* = x > 0$. Then

$$g(n_1^*, \ldots, n_k^*) = G(x) = \frac{n-x}{k-1} + (k-1)(d-1)\left(\frac{n-x}{k-1}\right)^{1-\frac{1}{d-1}} + x^{1-\frac{1}{d-1}}$$

$$G'(x) = -\frac{1}{k-1}$$

$$- (k-1)(d-1)\left(1-\frac{1}{d-1}\right)\left(\frac{1}{k-1}\right)\left(\frac{n-x}{k-1}\right)^{-\frac{1}{d-1}}$$

$$+ \left(1-\frac{1}{d-1}\right)x^{-\frac{1}{d-1}}$$

$$= -\frac{1}{k-1} - (d-2)\left(\frac{n-x}{k-1}\right)^{-\frac{1}{d-1}} + \frac{d-2}{d-1}x^{-\frac{1}{d-1}} < 0$$

since $\frac{n-x}{k-1} > x$. Furthermore, it is easily verified that $G''(x) < 0$. Hence $G(x)$ achieves its minimum at the boundary points $x = 0$ or $\frac{n-x}{k-1}$, and we either have Case (iia) or Case (i). The claim, hence Theorem 5, is proved.

5. An Application

A (rectangular) switch (see [1] for a general discussion) has the property that any set of pairs - one inlet and one outlet, can be simultaneously connected. In fact the fan-out property is also frequently assumed which allows the pairs to overlap. Consider a 2-stage network connecting a set of c channels to a set of u users. For the time being we assume that there is only one switch in the second stage and each first-stage switch has one outlet connected to one inlet of the second-stage switch. We are to assign channels to the inlets of the first-stage switches, and the users to the u outlets of the second-stage switch, such that any k channel-user pairs, for $k \leq u$, can be simultaneously connected. Determine r such that $u \leq F(r)$ (or $f(r)$). We assign each channel to r different first-stage switches by using some line systems in the d-dimensional space or in the plane and interpreting channels as nodes and first-stage switches as lines. Suppose that a set S of s channels has been requested. Then by the definition of $F(r)$ (or $f(r)$), any subset S' of S with s' users must have at least s' first-stage switches carrying them. Hence Hall's theorem on SDR (system of distinct representatives) [2] applies and we can find s distinct first-stage switches each carrying a distinct channel of S. Since each such first-stage switch can be connected to the second-stage switch independently, the simultaneous connection is done. Furthermore, if the fan-out property is assumed for the second stage switch, any channel may be simultaneously connected to any or all of the u users. This argument can be extended to a genuine two-stage network with m second-stage switches each having u outlets with $u \leq F(r)$ (or $f(r)$) and each first stage switch having m outlets, each connected to an inlet of a second stage switch. (See [4] for a more detailed account.)

The above discussion also makes it clear that the problem we studied can be interpreted as a generalized SDR problem for determining the conditions such that any k subsets have k distinct representatives (Hall's theorem deals with the case that k equals the cardinality of the given family of subsets). Our results give bounds on the number k when each subset has r elements.

References

[1] V.E. Benes, Mathematical Theory of Connecting Networks and Telephone Traffic, Academic Press, New York, 1965.

[2] P. Hall, "On representative of subsets", *Journal of the London Mathematical Society* 10 (1935) 26-30.

[3] G.H. Hardy, J.E. Littlewood, and G. Polya, Inequalities, Cambridge University Press, New York, 1934.

[4] G.W. Richards and F.K. Hwang, "A two-stage rearrangeable broadcast switching network", to appear.

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 165–182
Elsevier Science Publishers B.V. (North-Holland)

A Census of Orthogonal Steiner Triple Systems of Order 15

Peter B. Gibbons

Department of Computer Science
University of Auckland
Auckland
New Zealand

1. Preliminary definitions

A (v,b,r,k,λ)- *balanced incomplete block design (BIBD)* is an arrangement of v *elements* into b *blocks* such that: (i) each element appears in exactly r blocks; (ii) each block contains exactly k $(<v)$ elements; (iii) each pair of distinct elements appear together in exactly λ blocks. Well known necessary conditions for a (v,b,r,k,λ)-BIBD to exist are $vr = bk$ and $\lambda(v-1) = r(k-1)$. Because of this dependence we shall use the abbreviated notation (v,k,λ)-BIBD to denote a (v,b,r,k,λ)-BIBD. A (v,k,λ)-BIBD in which $v = b$ (and consequently $r = k$) is called *symmetric*.

Two (v,k,λ)-BIBD's D_1 and D_2, with element sets V_1 and V_2 respectively, are said to be *isomorphic* if there is a bijection $\theta: V_1 \to V_2$ such that $\{x_1, \ldots, x_k\}$ is a block of D_1 if and only if $\{\theta(x_1),\ldots,\theta(x_k)\}$ is a block of D_2. An *automorphism* of a BIBD is an isomorphism of the BIBD with itself. The set of all automorphisms, under the usual composition of mappings, forms the *automorphism group* of the BIBD.

A *Steiner triple system* of order v (STS(v)) is a $(v,3,1)$-BIBD. In this paper an STS(v) will often be represented by the pair (V,B), where V is the set of elements and B is the collection of blocks, or *triples*. Steiner triple systems have been studied extensively, and are known to exist if and only if the order $v \equiv 1, 3 \pmod 6$. For general references on STS's the reader is referred to Lindner and Rosa [6].

Two STS's (V,B_1) and (V,B_2) on the same set V are *disjoint* if $B_1 \cap B_2 = \phi$, i.e. if they have no triples in common. Furthermore, they are said to be *orthogonal* (or *perpendicular)* if they are disjoint and, moreover, satisfy the following property:

$$\{x,y,z\}, \{u,v,z\} \in B_1, \ \{x,y,a\}, \{u,v,b\} \in B_2 \to a \neq b$$

That is, if two pairs of elements appear with the same third element in triples of B_1, then they appear with distinct third elements in triples of B_2.

2. Background

Orthogonal STS's were first introduced by O'Shaughnessy [13] for the purpose of constructing Room squares. In his paper O'Shaughnessy displayed pairs of orthogonal STS's of orders 7, 13 and 19, and conjectured that such pairs exist for all orders $v \equiv 1 \ (mod\ 6)$. He also conjectured that pairs of orthogonal STS's do not exist for $v \equiv 3 \ (mod\ 6)$.

Mullin and Nemeth [9,10] supported these conjectures by showing that no orthogonal STS's of order 9 exist, and that there exists a pair of orthogonal STS's of order p^n, p a prime, for $p^n \equiv 1 \ (mod\ 6)$. However, the second conjecture of O'Shaughnessy was eventually disproved by Rosa [15] when he displayed a pair of orthogonal STS's of order 27.

Work has also been carried out by Gross [3,4] and Zhu [17] on constructing larger sets of mutually orthogonal STS's. Gross, for example, has constructed a set of 6 mutually orthogonal STS's of order 31, the smallest order for which more than two mutually orthogonal STS's are known to exist.

In other work Mullin and Rosa [11], Zhu [18], Mendelsohn [8], Mullin and Vanstone [12], and Lawless [5] have extended the concept of orthogonality for STS's to Steiner systems in general, and have investigated applications, for example, to the construction of generalised Room squares.

Finally, a brief survey of known results on orthogonal STS's is included in Rosa [14]. In this paper Rosa mentions that the smallest orders $v \equiv 3 \ (mod\ 6)$ for which it is undecided whether a pair of orthogonal STS(v)'s exist are $v =$ 15, 21, 33, 39, 45, 51, 63, 69, and 75. Furthermore, no work has currently been undertaken on enumerating such pairs for $v \geq 13$.

In this paper we confirm that there exists only one pair of orthogonal STS(13)'s. Furthermore we establish the existence of exactly 19 non-equivalent pairs of orthogonal STS(15)'s involving 24 non-isomorphic systems.

3. Method

The enumeration was carried out on a computer (or, more specifically, a set of four Hewlett Packard 9836 microcomputers based on the Motorola MC68000 16-bit microprocessor) using construction and enumeration techniques adapted from Gibbons, Mathon and Corneil [1,2]. Given an input *basis* STS a backtrack search strategy is used to attempt to construct an orthogonal *mate*. The construction proceeds block by block, in lexicographical order, subject both to the normal constraints of an STS, and also to the orthogonality constraints imposed by the basis STS. On detection of a violation of these constraints, the program must backtrack. On completion of an orthogonal mate, the program continues to search for further mates.

The program was coded in UCSD Pascal on the HP micros and tested with Steiner triple systems of small orders. It easily found the known pair of orthogonal STS(7)'s, and was also quick to confirm the result of Mullin and Nemeth [10] concerning the non-existence of a pair of orthogonal STS(9)'s.

In the case of the STS(13)'s, there are two (non-isomorphic) systems to consider. In about 30 minutes of CPU time for each basis design the program was able to establish that the transitive STS(13) (#1 in Mathon, Phelps and Rosa [7]) has exactly one mate, isomorphic to itself, whereas the other STS(13) has no mate. The orthogonal pair is listed in the Appendices.

The first real test for the program came with the case of the STS(15)'s. It is well known that there are exactly 80 non-isomorphic such systems (White, Cole and Cummings [16]). A more recent listing of these systems together with a comprehensive summary of their properties may be found in Mathon, Phelps and Rosa [7]. Indeed, all numberings and representations of basis designs (unless otherwise stated) conform to this reference.

It was not surprising that computation times for the STS(15)'s were significantly greater, and, after a couple of systems had been examined (without detection of a mate) it was realised that more sophisticated techniques would have to be utilised if a complete enumeration was to be contemplated. The most obvious technique to apply was that of *isomorph rejection.*

Suppose we have two pairs $\{D_1,D_1'\}$, $\{D_2,D_2'\}$ of orthogonal STS(v)'s on the same treatment set V. Each pair forms a (v,3,2)-BIBD (or *twofold triple system),* and we say that the pairs are *equivalent* if the corresponding twofold triple systems are isomorphic. Furthermore, we say that the pairs are *isomorphic* if there is an isomorphism ϕ: $V \to V$ mapping D_1 to D_2 and D_1' to D_2'. It is apparent that isomorphism implies equivalence of pairs, but in general it is not known whether the converse is true. However, in the case of the STS(15)'s "equivalence" *is* equivalent to "isomorphism" - we found no pairs of orthogonal STS(15)'s which are equivalent but *not* isomorphic.

Clearly, in our search process we would like to avoid generating isomorphic pairs of orthogonal STS's involving a common basis design. Given a particular basis design D_1, we shall be generating pairs of the form $\{D_1,D_1'\}$, $\{D_1,D_1''\}$. Suppose G is the automorphism group of D_1. Then a special type of isomorphism occurs if there exists a $\phi \in G$ such that $D_1'' = \phi(D_1')$. This fact can be used to implement the following isomorph rejection procedure.

Suppose we have constructed a partial orthogonal system D_2' to a given basis design D_1 with automorphism group G. Now, such partial systems are being considered by the backtrack program in increasing lexicographical order, so that if there exists a $\phi \in G$ such that $\phi(D_2') < D_2'$, then we can *reject* D_2', since an equivalent partial system has already been considered earlier in the search. For greatest effect this check should, in theory, be applied after the completion of each new block in the constructed (partial) mate. In practice, however, this would be too costly, so instead we opted to apply the check, in the case of the STS(15)'s, only after construction of each of the first 7 blocks, viz. those blocks in the constructed mate containing the element 1.

We now observe that isomorph rejections will largely be effected by group elements from the stabilizer of G which fixes the element 1. For the isomorph rejection procedure to have the greatest effect, it would seem that the basis design should be in a form which maximizes the size of this stabilizer. In many cases a transformation from the representation given in [7] is necessary to accomplish this. For example, take system #31 in [7] which has an automorphism group of order 4 with generators

(1 5 6 15) (2 3 8 13) (4 11 9 14) (7 12).

Here the stabilizer fixing 1 contains only the identity mapping. However if we interchange elements 1 and 10 in this design, the stabilizer becomes the automorphism group itself, of order 4.

This isomorph rejection procedure, once implemented, resulted in a considerable improvement in search efficiency. For example, in the case of the previously mentioned STS(13)'s, the search time for each basis design was cut to about 10 minutes. In the case of the STS(15)'s 44 of the 80 systems have non-trivial groups, and in all but 3 or 4 of these cases a representation can be obtained with a non-trivial 1-stabilizer. The best example, of course, is system #1 which is 2-transitive with a group of order 20,160. Equivalence considerations here imply that only {1,2,4} needs to be considered for block 1 of the mate, while the only non-equivalent possibilities for block 2 are {1,3,5}, {1,3,6}, {1,3,7}, and {1,3,8}. For this particular case the search time was about 3 hours. On the other hand, a few systems, such as #75, #76, and #77, could not be transformed to give a non-trivial 1-stabilizer. In these cases we tried applying the isomorph rejection check to later blocks, but found that the extra cost outweighed any benefit obtained from rejection of equivalent partial systems.

The remaining 36 (rigid) systems with trivial automorphism groups proved to be difficult cases to check, accounting for about 75% of the total search time. Having checked all 44 systems with non-trivial groups we searched hard for a good heuristic to assist with these cases. One observation which we thought might help was the fact that none of these 36 rigid systems have a subsystem of order 7. If we could restrict ourselves to constructing mates with no order-7 subsystems then we would effectively avoid generating a large proportion of mates which had already been examined as basis systems. Unfortunately however subsystems can only be found in the constructed mate after at least 15 blocks have been constructed. This proved to be too late to be cost effective in the search.

4. Results

A number of pairs of orthogonal STS(15)'s were found using the search procedure described in the previous section. We display these pairs in the form of a multi-graph, where the vertices represent the set of 80 STS(15)'s, and there is an edge between each distinct pair of orthogonal systems. Note that our graph may contain self-loops.

The complete set of connected components of this graph (omitting isolated vertices with no self-loops) is displayed in Figure 1. Beside each node in this figure we have indicated the order of the automorphism group of the corresponding STS. With one type of exception (described below), Figure 1 represents all distinct pairs of orthogonal STS(15)'s relative to our chosen representations for the basis designs. Some of the pairs are isomorphic, viz. those corresponding to starred edges with common end-points. Using isomorphism checking procedures developed in [1,2] we have established that these are the *only* isomorphisms, or equivalences for that matter. In other words, repeating what we mentioned earlier, we found no pairs of orthogonal STS(15)'s that are equivalent but *not* isomorphic.

The actual representations of systems making up the listed components are contained in the Appendices. We note here that the isomorphisms indicated in components 2, 5 and 7 are similar in form. If we denote a typical pair by $\{basis, mate\ \#1\}$, $\{basis, mate\ \#2\}$, then the isomorphism maps $basis \to mate\ \#2$ and $mate\ \#1 \to basis$. The respective isomorphisms are listed in the Appendices.

Another type of isomorphism is not displayed in Figure 1. Note that all designs in the figure have groups of order 1 or 3. In particular, given an orthogonal pair $\{D_1, D_2\}$ where D_1 and D_2 have groups of order 3 and 1 respectively, we can obtain two additional isomorphic, but distinct pairs by

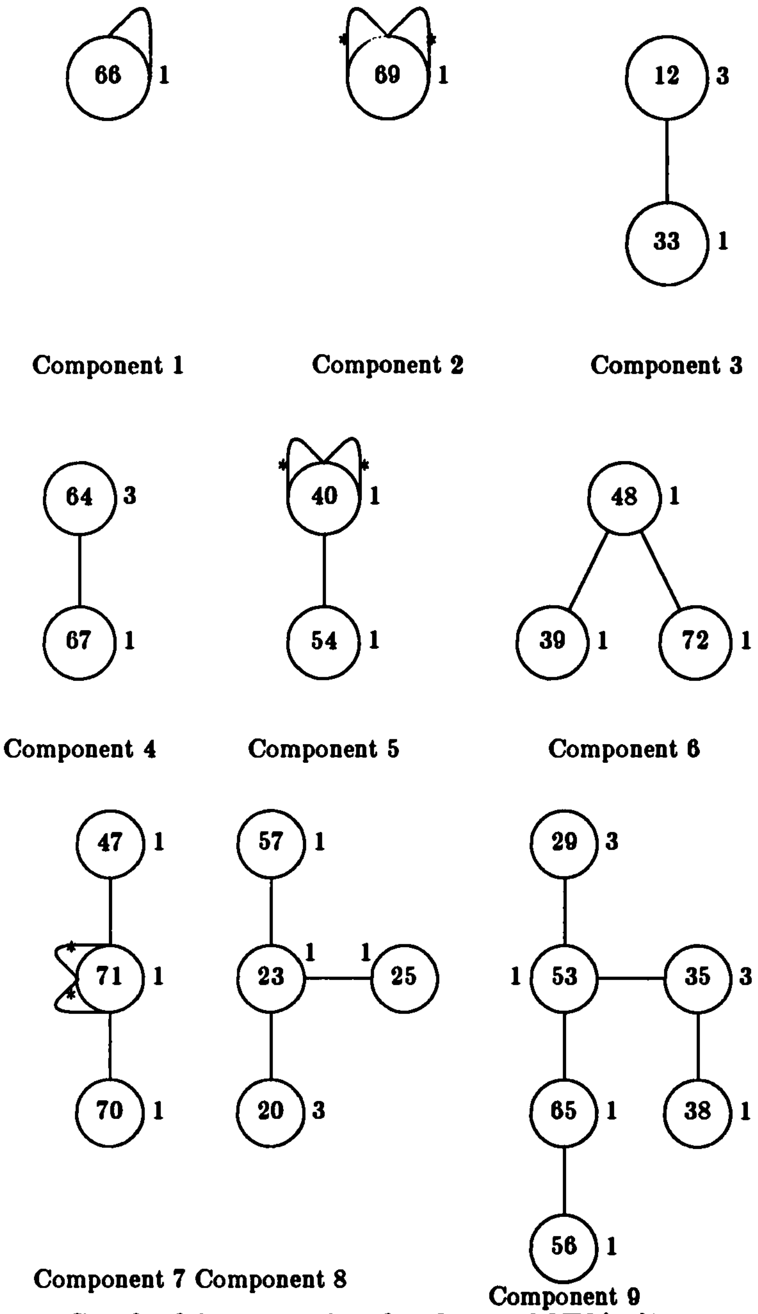


Figure 1 : Graph of distinct pairs of orthogonal STS(15)'s

applying the group of D_1 to the pair. Pairs obtained by this process have not been displayed either in Figure 1 or the Appendices.

5. Remarks

Surveying the properties of the systems in the above components, it is difficult to identify any trends which might be helpful in constructing orthogonal systems of higher orders. It is interesting that all systems have small groups with most being rigid, which accounts for the difficulty in finding such systems. Also, only two systems, #12 and #20, contain a subsystem of order 7.

Another observation is the following. Consider a pair $D_1 = (V,B_1)$, $D_2 = (V,B_2)$ of orthogonal STS(v)'s, and, for any $x \in V$, define $P_x = \{\{y,z\}: \{x,y,z\} \in B_1\}$ as the set of element pairs occurring with x in a block of D_1. Now define $Q_x = \{w: \{w,y,z\} \in B_2, \{y,z\} \in P_x\}$ as the set of elements occurring with pairs of P_x in blocks of B_2. Then $R = \{Q_x: x \in V\}$ is a 1-design with v blocks, each containing $(v-1)/2$ elements, and with each element occuring in $(v-1)/2$ blocks. If $v \equiv 3 \pmod 4$ the parameters are admissible for R to form a 2-design, viz. a symmetric $(v,(v-1)/2,(v-3)/4)$−BIBD (or *Hadamard* design).

For example with the (unique) pair of orthogonal STS(7)'s R forms a symmetric (7,3,1)-BIBD (or STS(7)). The next admissible order for R to be a Hadamard design is $v = 15$. However analysis of all generated pairs of orthogonal pairs of STS(15)'s reveals that none induces a symmetric (15,7,3)-BIBD.

It would be interesting to determine the admissible orders $v > 15$ for which such Hadamard designs are formed. In particular it would be of interest to know whether there are any orthogonal STS(19)'s which induce symmetric (19,9,4)-BIBD's. The known pair, generated in [13] and listed in the Appendices, does not induce such a design.

We also note that there is no set of three mutually orthogonal STS(15)'s. An open question is to determine the smallest order for which there exists a set of more than two mutually orthogonal systems. As mentioned earlier, currently the smallest known order for this to occur is $v = 31$.

Finally we remark that no pair of orthogonal STS(21)'s have yet been found. We are convinced that such a pair exists. However this case was beyond the range of the computational techniques described in this paper.

Acknowledgements

The author is grateful to Alex Rosa for initially suggesting that this problem might be computationally tractable, and for also providing advice and encouragement as the search progressed. The author is also indebted to Rudi Mathon for sharing his enthusiasm and considerable experience with problems of this type. Finally the author would like to thank the referee, whose comments were precise and positive. This work was carried out while the author was visiting at the University of Toronto, Canada.

References

[1] P.B. Gibbons, *Computing techniques for the construction and analysis of block designs* Technical Report #92, Department of Computer Science, University of Toronto (May 1976).

[2] P.B. Gibbons, R.A. Mathon, and D.G. Corneil, "Computing techniques for the construction and analysis of block designs", *Utilitas Mathematica 11 (1977) 161-192.*

[3] K.B. Gross, "On the maximal number of orthogonal Steiner triple systems", *Proceedings of the Conference on Algebraic Aspects of Combinatorics* (Toronto 1975), Congressus Numerantium XIII, Utilitas Mathematica Publishing Inc., Winnipeg 1975, 233-240.

[4] K.B. Gross, "On the maximal number of pairwise orthogonal Steiner triple systems", *Journal of Combinatorial Theory (A)* 19 (1975), 256-263.

[5] J.F. Lawless, "Pairwise balanced designs and the construction of certain combinatorial systems", *Proceedings of the Second Louisiana Conference on Combinatorics, Graph Theory and Computing* (Baton Rouge 1971), 353-366.

[6] C.C. Lindner and A. Rosa, *Topics on Steiner Systems, Annals of Discrete Mathematics* 7 (1980).

[7] R.A. Mathon, K.T. Phelps, and A. Rosa, "Small Steiner triple systems and their properties", *Ars Combinatoria* 15 (1983) 3-110.

[8] N.S. Mendelsohn, "Orthogonal Steiner systems", *Aequationes Mathematicae* 5 (1970) 268-272.

[9] R.C. Mullin and E. Nemeth, "On furnishing Room squares", *Journal of Combinatorial Theory* 7 (1969) 266-272.

[10] R.C. Mullin and E. Nemeth, "On the non-existence of orthogonal Steiner systems of order 9", *Canadian Mathematical Bulletin* 13 (1970) 131-134.

[11] R.C. Mullin and A. Rosa, "Orthogonal Steiner systems and generalized Room squares", *Proceedings of the Sixth Manitoba Conference on Numerical Mathematics* (Winnipeg 1976), Congressus Numerantium XVIII, Utilitas Mathematica Publishing Inc., Winnipeg 1977, 315-323.

[12] R.C. Mullin and S.A. Vanstone, "Steiner systems and Room squares", in *Topics on Steiner systems* (C.C. Lindner and A. Rosa; editors), *Annals of Discrete Mathematics* 7 (1980) 95-104.

[13] C.D. O'Shaughnessy, "A Room design of order 14", *Canadian Mathematical Bulletin* 11 (1968) 191-194.

[14] A. Rosa, "Intersection properties of Steiner systems", in *Topics on Steiner systems* (C.C. Lindner and A. Rosa; editors), *Annals of Discrete Mathematics* 7 (1980) 115-128.

[15] A. Rosa, "On the falsity of a conjecture on orthogonal Steiner triple systems", *Journal of Combinatorial Theory (A)* 16 (1974) 126-128.

[16] H.S. White, F.N. Cole, and L.D. Cummings, "Complete classification of the triad systems on fifteen elements", *Memoirs of the National Academy of Sciences U.S.A.* 14, 2nd. memoir (1919), 1-89.

[17] L. Zhu, "A construction for orthogonal Steiner triple systems", *Ars Combinatoria* 9 (1980) 253-262.

[18] L. Zhu, "A pair of orthogonal Steiner systems S(3,5,17)", *Discrete Mathematics* 31 (1980) 111-113.

Appendices

A1. The pair of orthogonal STS(7)'s

#1

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

Isomorphism to mate: (3 7 5 4)

A2. The pair of orthogonal STS(13)'s

#1

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

Isomorphism to mate: (2 3 5 4 9 8) (6 10 11 12 13 7)

A3. The pairs of orthogonal STS(15)'s

Component 1

#66 (basis)

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

Isomorphism of basis to mate: (1 11) (3 13) (4 12) (5 10) (6 14) (8 9)

Component 2

#69 (basis)

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

Isomorphism of basis to mate #1: (1 11 15 9 5 7 6 2 8 10 12 3 14)

Isomorphism of basis to mate #2: (1 14 3 12 10 8 2 6 7 5 9 15 11)

Isomorphism {basis,mate #1} → {basis,mate #2}:

(1 14 3 12 10 8 2 6 7 5 9 15 11)

Component 3

#12 (basis)

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

#33 (mate)

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

Component 4

#64 (basis)

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

#67 (mate)

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

Component 5

#40 (basis)

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

#54 (mate #1)

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

Isomorphism of basis to mate #2: (1 9 8 10) (2 4 11 13 12 5 3 15 6)

Isomorphism of basis to mate #3: (1 10 8 9) (2 6 15 3 5 12 13 11 4)

Isomorphism {basis,mate #2} → {basis,mate #3}:

(1 10 8 9) (2 6 15 3 5 12 13 11 4)

Component 6

#48 (basis)

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

#39 (mate #1)

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

#72 (mate #2)

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

Component 7

#71 (basis)

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

#47 (mate #1)

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

#70 (mate #2)

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

Isomorphism of basis to mate #3: (1 9 6 15 2 11 3) (4 7 14 5 10 13) (8 12)

Isomorphism of basis to mate #4: (1 3 11 2 15 6 9) (4 13 10 5 14 7) (8 12)

Isomorphism {basis,mate #3} → {basis,mate #4}:

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

Component 8

#23 (basis)

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

#20 (mate #1)

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

#25 (mate #2)

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

#57 (mate #3)

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

Component 9

#53 (basis)

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

#29 (mate #1)

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

#35 (mate #2)

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

#65 (mate #3)

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

#38 (mate to #35)

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

#56 (mate to #65)

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

A4. The known pair of orthogonal STS(19)'s

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

Isomorphism to mate: (2 5 17 8 10 18 12 7 6) (3 9 14 15 19 16 4 13 11)

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 183–208

Derived Steiner Triple Systems of Order 15

M.J. Grannell and T.S. Griggs

Division of Mathematics,
Preston Polytechnic,
Corporation Street,
Preston, PR1 2TQ,
Lancashire, England.

1. Introduction

Denote a Steiner system by $S(t,k,v)$ where the parameters have their usual meaning. It is an elementary proposition that if any point of a Steiner system is chosen, all blocks not containing the point are deleted, and the point itself is then deleted from all of the remaining blocks, what remains is another Steiner system $S(t-1,k-1,v-1)$. The latter system is said to be derived from the former. It is well known that necessary and sufficient conditions are as follows: for a Steiner triple system $S(2,3,v)$ or $STS(v)$, $v \equiv 1$ or $3 \pmod 6$ while for a Steiner quadruple system $S(3,4,v)$ or $SQS(v)$, $v \equiv 2$ or $4 \pmod 6$. Such v are called admissible. It follows that there exists a derived Steiner triple system for every admissible order. However, whether or not every Steiner triple system is derived is a fascinating open question.

For $v = 7$ and 9, the Steiner triple systems are unique up to a isomorphism and are therefore derived. The case when $v = 13$ was solved by Mendelsohn and Hung [7] who showed that both of the two non-isomorphic systems which exist for this order are also derived. There are 80 non-isomorphic Steiner triple systems of order 15 (see [2] and [4]). In this paper we shall use the listing of these given by Bussemaker and Seidel [1], and also given in [5] where it is probably more easily accessible. The present state of knowledge concerning the derivability of these systems is given in the survey paper by Phelps [10]. It rests heavily on general theorems, also by Phelps, in earlier papers [8], [9]. In the first of these he proves:

Theorem A (Phelps [8])

A Steiner triple system of order $2v+1$ with a derived Steiner triple system of order v is itself derived.
This theorem shows that 23 of the 80 systems, namely #1-22 and 61, are derived since they contain the $STS(7)$, {1,2,3}, {1,4,5}, {1,6,7}, {2,4,6}, {2,5,7}, {3,4,7}, {3,5,6}.

In the second paper a theorem equivalent to the following is proved:

Theorem B (Phelps [9])

If a Steiner triple system of order $2v+1$ contains all but one of the blocks of a Steiner triple system of order v, and this $STS(v)$ is derived then the $STS(2v+1)$ is also derived. (In [2] an $STS(v)$ with one block missing is called a semi-head).

This theorem shows that 15 more systems, namely #23-34, 62, 63 and 64 are derived since they contain the semi-head {1,2,3}, {1,4,5}, {1,6,7}, {2,4,6}, {2,5,7}, {3,4,7}.

Finally in [10], Phelps states that he has himself determined that #35 and 53 are derived and that Gibbons [3] has added #59, 70 and 76. The $SQS(16)$'s containing these $STS(15)$'s as derived systems are exhibited in the recent encyclopaedic paper by Mathon, Phelps and Rosa [6]. Thus the total number of known derived $STS(15)$'s is 43. In this paper we raise this number to 66.

2. Methodology

Our general methodology is an extension of that used by Phelps [8], [9], in the proof of his theorems quoted above. Our method is applicable only to Steiner triple systems of order 15 and involves the use of a computer search. We analyse the situation in which an $STS(15)$ contains an $STS(7)$ apart from two blocks (a demi-semi-head?). First we need a definition.

Definition. A quadrilateral consists of four blocks of a Steiner triple system whose union has cardinality six.

It is clear that a quadrilateral must have the following configuration: $\{a,b,c\}$, $\{a,y,z\}$, $\{x,b,z\}$, $\{x,y,c\}$. When such a collection appears in a Steiner triple system it may be removed and replaced by the "opposite" quadrilateral $\{x,y,z\}$,$\{x,b,c\}$, $\{a,y,c\}$, $\{a,b,z\}$ to form a different (but possibly isomorphic) Steiner triple system. Gibbons [3] has shown that precisely 79 of the 80 $STS(15)$'s contain at least one quadrilateral and that these may be transformed into one another by repeated changing of quadrilaterals as described.

Note firstly that the inclusion of a quadrilateral within an $STS(15)$ is equivalent to the $STS(15)$ containing five of the seven blocks of an $STS(7)$. We now proceed with the analysis.

Let the quadrilateral be $\{a_1,a_3,a_5\}$, $\{a_1,a_4,a_6\}$, $\{a_2,a_3,a_6\}$, $\{a_2,a_4,a_5\}$. Identify the three pairs of elements which are not included in the quadrilateral and list the three blocks of the *STS*(15) which contain these pairs. Suppose these are $\{a_1,a_2,x\}$, $\{a_3,a_4,y\}$, $\{a_5,a_6,z\}$. Then none of x, y and z can equal any a_i and, moreover, we can assume that x,y and z are themselves unequal (for otherwise the *STS*(15) would contain either an *STS*(7) or a semi-head which can be dealt with by Phelps' theorems). Select one of these latter three blocks. Without loss of generality we will choose $\{a_1,a_2,x\}$. Next identify the blocks which contain the pairs $\{a_3,x\}$, $\{a_4,x\}$, $\{a_5,x\}$, $\{a_6,x\}$. Let these be $\{a_3,x,b_3\}$, $\{a_4,x,b_4\}$, $\{a_5,x,b_5\}$, $\{a_6,x,b_6\}$. The b_i's must be distinct from one another and from each of the a_i's. Also, $y \neq b_3$ or b_4 and $z \neq b_5$ or b_6. However, it is possible for y to be equal to b_5,b_6 or z to be equal to b_3,b_4 (but not simultaneously). The above blocks are 11 of the 35 blocks in an *STS*(15).

Since each element occurs 7 times within an *STS*(15), there are in addition four more blocks containing a_1 and likewise for a_2, three more blocks containing a_3 and likewise for a_4, a_5 and a_6, and two more blocks containing x, all these blocks being distinct and numbering 22 in all. It is left to identify the remaining two blocks. A counting argument shows that these contain the 'six' elements y, z, b_3, b_4, b_5, b_6. It is to be understood that if, for example, $y = b_5$ then this element appears twice, that is once in each of the two blocks. The exact partition of the elements into the two blocks is not determined. We now make the further assumption that these two blocks are $\{b_3, b_4, y\}$ and $\{b_5, b_6, z\}$ i.e. that the configuration of the *STS*(15) is as given below.

$\{a_1,a_3,a_5\}$, $\{a_1,a_4,a_6\}$, $\{a_2,a_3,a_6\}$, $\{a_2,a_4,a_5\}$,

$\{a_1,a_2,x\}$, $\{a_3,a_4,y\}A$, $\{a_5,a_6,z\}B$,

$\{a_3,x,b_3\}A$, $\{a_4,x,b_4\}A$, $\{a_5,x,b_5\}B$, $\{a_6,x,b_6\}B$,

$\{b_3,b_4,y\}A$, $\{b_5,b_6,z\}B$, together with the 22 blocks identified above.

The four blocks labelled A form a quadrilateral as do the four labelled B. Replacing these with the "opposite" quadrilaterals gives the following transformed *STS*(15).

$$\{a_1,a_3,a_5\},\quad \{a_1,a_4,a_6\},\quad \{a_2,a_3,a_6\},\quad \{a_1,a_4,a_5\},$$

$$\{a_1,a_2,x\ \},\quad \{a_3,a_4,x\ \}A,\ \{a_5,a_6,x\ \}B,$$

$$\{a_1,\qquad\},\quad \{a_1,\qquad\},\quad \{a_1,\qquad\},\quad \{a_1,\qquad\},$$

$$\{a_2,\qquad\},\quad \{a_2,\qquad\},\quad \{a_2,\qquad\},\quad \{a_2,\qquad\},$$

$$\{a_3,b_3,y\ \}A,\ \{a_3,\qquad\},\quad \{a_3,\qquad\},\quad \{a_3,\qquad\},$$

$$\{a_4,b_4,y\ \}A,\ \{a_4,\qquad\},\quad \{a_4,\qquad\},\quad \{a_4,\qquad\},$$

$$\{a_5,b_5,z\ \}B,\ \{a_5,\qquad\},\quad \{a_5,\qquad\},\quad \{a_5,\qquad\},$$

$$\{a_6,b_6,z\ \}B,\ \{a_6,\qquad\},\quad \{a_6,\qquad\},\quad \{a_6,\qquad\},$$

$$\{x,b_3,b_4\ \}A,\ \{x,b_5,b_6\ \}B,\ \{x,\qquad\},\quad \{x,\qquad\}.$$

This latter $STS(15)$ contains an $STS(7)$ on the elements a_1,a_2,a_3,a_4,a_5,a_6, and x and hence may be extended to an $SQS(16)$ using Phelps' techniques. The method is as follows:

(1) The $STS(7)$ is extended to an $SQS(8)$ with one extra element, say ∞.

(2) The other 28 blocks of the $STS(15)$ all have the element ∞ adjoined to them.

(3) Another $SQS(8)$ is formed on the elements b_3,b_4,b_5,b_6,y,z, and two further elements b_1 and b_2.

(4) A one-factorization of a graph K_8 whose vertices are the elements a_1,a_2,a_3,a_4,a_5,a_6, x, and ∞ is formed. The system $SQS(16)$ is then completed by taking each edge $\{a_i,a_j\}$, $i\neq j$ or $\{a_i,x\}$ in turn and identifying the edge $\{\infty,a_k\}$ or $\{\infty,x\}$ within the same one-factor. The element a_k or x occurs four times in blocks of the $STS(15)$ with disjoint pairs of elements from the set $\{b_1,b_2,b_3,b_4,b_5,b_6, y, z\}$. Four new blocks are formed each containing one of these pairs together with $\{a_i,a_j\}$, $i\neq j$ or $\{a_i,x\}$.

Clearly stages (3) and (4) contain some flexibility. In carrying out these steps, it may be possible to arrange that the four 3-blocks in each of the two quadrilaterals (A & B) of the original, untransformed $STS(15)$ receive the same fourth element in the $SQS(16)$. It is then possible to transform the $SQS(16)$ to another $SQS(16)$ containing the original $STS(15)$ as a derived subsystem.

Our method was, therefore, to search each $STS(15)$ for quadrilaterals and determine which of these extend to the configuration described. This we did by computer. Using the configuration to extend the $STS(15)$ to an $SQS(16)$ was then undertaken by hand and was found to be a not too onerous task.

3. Results

In searching for the configuration described in the previous section, the computer results indicated that in addition to the 15 systems identified in [9], 9 further $STS(15)$'s, including both of the additional systems considered by Phelps [10] and one of the three considered by Gibbons [3] have derived semi-heads. Hence it follows from theorem B that these systems are derived.

The systems, together with their semi-heads, are:

#35,	{1,4,5},	{1,10,11},	{1,12,13},	{4,11,13},	{5,10,13},	{5,11,12}
#39,	{1,6,7},	{1,8,9},	{6,8,15},	{6,9,13},	{7,8,13},	{7,9,15}
#40,	{1,4,5},	{1,10,11},	{1,14,15},	{4,10,15},	{4,11,14},	{5,10,14}
#41,	{1,6,7},	{1,8,9},	{6,8,15},	{6,9,13},	{7,8,13},	{7,9,15}
#47,	{1,6,7},	{1,8,9},	{1,14,15},	{6,8,14},	{6,9,15},	{7,9,14}
#53,	{2,4,6},	{2,13,15},	{4,10,15},	{4,11,13},	{6,10,13},	{6,11,15}
#54,	{2,4,6},	{2,9,11},	{4,9,15},	{4,11,14},	{6,9,14},	{6,11,15}
#58,	{1,10,11},	{1,14,15},	{4,10,15},	{4,11,14},	{6,10,14},	{6,11,15}
#59,	{1,6,7},	{1,8,9},	{1,14,15},	{6,9,15},	{7,8,15},	{7,9,14}

Apart from these, 19 $STS(15)$'s (including the other two considered by Gibbons), contain the configuration described in the previous section. Using the configuration we could find in each case an $SQS(16)$ with the $STS(15)$ as a derived system. These $SQS(16)$'s are given in the Appendix; the 35 blocks of each $STS(15)$ all have a further element (16) adjoined to them and these blocks are listed down the first column. Thus the $STS(15)$'s may be checked against the listings in [1] or [5] by the reader. The $SQS(16)$'s have been checked by the authors using a computer checking program.

The situation concerning derived $STS(15)$'s is now as follows:

1. 23 systems contain an $STS(7)$ and are thus derived by theorem A. These are #1-22 and 61.
2. 24 systems contain a semi-head and are thus derived by theorem B. These are #23-35, 39, 40, 41, 47, 53, 54, 58, 59, 62, 63 and 64.
3. 19 systems contain the configuration described in this paper and are derived as indicated in the Appendix.
   These are #36, 38, 43-46, 48-52, 55, 56, 57, 60, 70, 74, 75 and 76.
4. 14 systems remain whose derivability is still undetermined.
   These are #37, 42, 65-69, 71, 72, 73 and 77-80.

References

[1] F. C. Bussemaker and J. J. Seidel, "Symmetric Hadamard matrices of order 36", Technological University Eindhoven, Report 70 WSK-02 (1970).

[2] F. N. Cole, L. D. Cummings and H. S. White, "Complete classification of the triad systems on fifteen elements", *Memoirs of the National Academy of Sciences U.S.A. 14*, Second memoir (1919), 1-89.

[3] P. B. Gibbons, "Computing techniques for the construction and analysis of block designs", Ph.D. Thesis, University of Toronto (1976). (also Department of Computer Science, University of Toronto, Technical Report No. 92, May 1976).

[4] M. Hall, Jr. and J. D. Swift, "Determination of Steiner triple systems of order 15", *Mathematical Tables and Other Aids to Computation* 9 (1955) 146-152.

[5] M. Limbos, "Projective embeddings of small Steiner triple systems", *Annals of Discrete Mathematics* 7 (1980) 151-173.

[6] R. A. Mathon, K. T. Phelps and A. Rosa, "Small Steiner triple systems and their properties", *Ars Combinatoria* 15 (1983) 3-110.

[7] N. S. Mendelsohn and S. H. Y. Hung, "On the Steiner systems S(3,4,14) and S(4,5,15)", *Utilitas Mathematica* 1 (1972) 5-95.

[8] K. T. Phelps, "Some sufficient conditions for a Steiner triple system to be a derived triple system", *Journal of Combinatorial Theory* (A) 20 (1976) 393-397.

[9] K. T. Phelps, "Some derived Steiner triple systems", *Discrete Mathematics* 16 (1976) 343-352.

[10] K. T. Phelps, "A survey of derived triple systems", *Annals of Discrete Mathematics* 7 (1980) 105-114.

APPENDIX

SYSTEM NUMBER 36

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

SYSTEM NUMBER 38

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

SYSTEM NUMBER 43

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

SYSTEM NUMBER 44

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

SYSTEM NUMBER 45

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

SYSTEM NUMBER 46

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

SYSTEM NUMBER 48

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

SYSTEM NUMBER 49

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

SYSTEM NUMBER 50

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

SYSTEM NUMBER 51

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

SYSTEM NUMBER 52

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

SYSTEM NUMBER 55

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

SYSTEM NUMBER 56

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

SYSTEM NUMBER 57

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

SYSTEM NUMBER 60

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

SYSTEM NUMBER 70

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

SYSTEM NUMBER 74

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

SYSTEM NUMBER 75

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

SYSTEM NUMBER 76

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

Annals of Discrete Mathematics 26 (1985) 209–220

A Survey of Results on the Number of $t-(v,k,\lambda)$ Designs

Hans-Dietrich O.F. Gronau

Wilhelm-Pieck-Universität
Sektion Mathematik
2500 Rostock
German Democratic Republic

A $t-(v,k,\lambda)$ design is a system of (not necessarily distinct) k-element subsets (called blocks) of a v-element set K such that every t-element subset of K appears exactly λ times in the blocks. Two $t-(v,k,\lambda)$ designs M and N are called *isomorphic* if and only if there is a permutation of the elements of K which bijectively transforms M into N. A $t-(v,k,\lambda)$ design B is called *indecomposable* (or elementary) if and only if there is no subsystem B' of B which is a $t-(v,k,\lambda')$ design for $0<\lambda'<\lambda$. The existence of a $t-(v,k,\lambda)$ design implies that

$$\frac{\lambda\binom{v-i}{t-i}}{\binom{k-i}{t-i}}$$

is an integer for every $i = 0,1,\ldots,t-1$.

We introduce the following notations:

$f(v,k,t,\lambda)$
: is the number of pairwise nonisomorphic $t-(v,k,\lambda)$ designs without repeated blocks.

$f^*(v,k,t,\lambda)$
: is the number of pairwise nonisomorphic indecomposable $t-(v,k,\lambda)$ designs without repeated blocks.

$g(v,k,t,\lambda)$
: is the number of pairwise nonisomorphic $t-(v,k,\lambda)$ designs with repeated blocks.

$g^*(v,k,t,\lambda)$
: is the number of pairwise nonisomorphic indecomposable $t-(v,k,\lambda)$ designs with repeated blocks.

For Steiner systems, i.e. $t-(v,k,1)$ designs, we refer the reader to the excellent bibliography and survey of Doyen and Rosa [D5]. In this paper, we survey results on these four functions, especially for $\lambda \geq 2$.

The determination of these functions is of practical interest but successfully attackable only for small parameters. In table 1, we present results on designs without repeated blocks for $6 \leq v \leq 9$ and all possible λ according to (1). For $v \geq 10$ results are only known for the smallest possible λ. In table 2 we give results for $10 \leq v \leq 16$ and all possible pairs (t,k), $2 \leq t < k \leq \frac{v}{2}$ and the smallest possible λ with respect to (1). Note that $f(v,k,t,\lambda) = f^*(v,k,t,\lambda)$ for these parameters. Finally in table 3 we present results for designs with repeated blocks. Since Steiner systems cannot have repeated blocks, we omit those parameter sets here.

In the literature results on this topic are published in many different journals and are rediscovered more often than they are discovered. Therefore a general improved communication is desirable. The author hopes that this paper contributes to this aim. I apologize to those whose results are left out here, if there are any. I should greatly appreciate reprints, preprints, and other information on further results.

Table 1. Designs without repeated blocks, $6 \le v \le 9$.				
$t-(v,k,\lambda)$	$f(v,k,t,\lambda)$	Reference	$f^*(v,k,t,\lambda)$	Reference
2-(6,3,2)	1	N2,R1	1	
2-(6,3,4)	1		0	
2-(7,3,1)	1		1	
2-(7,3,2)	1	N1,G4,R2,I1	0	K1,G4,R2
2-(7,3,3)	1	M4,G4,R2,I1	1	R1,G4
2-(7,3,4)	1		0	R1
2-(7,3,5)	1		0	
2-(8,3,6)	1		1	
2-(8,4,3)	4	N1,G2,G9,I1	4	
2-(8,4,6)	164	G11	128	G12
2-(8,4,9)	164	G11	1	G12
2-(8,4,12)	4	G9	0	G9
2-(8,4,15)	1		0	
3-(8,4,1)	1	B1,M3,G8	1	
3-(8,4,2)	1	M3,G8,I1	0	G8
3-(8,4,3)	1	M3,G8,I1	1	G8
3-(8,4,4)	1	G8	0	G8
3-(8,4,5)	1		0	
2-(9,3,1)	1		1	
2-(9,3,2)	13	M1,G2,H5,I1	11	M1,H5
2-(9,3,3)	330	I1,H6†	171	H6†
2-(9,3,4)	330	I1,H6†	0	H6†
2-(9,3,5)	13	H5	0	H5
2-(9,3,6)	1	H5	0	H5
2-(9,3,7)	1		0	
2-(9,4,3)	11	L2,B4,G2,I1	11	
2-(9,4,3q),$2 \le q \le 5$	?		?	
2-(9,4,18)	11		?	
2-(9,4,21)	1		0	
3-(9,4,6)	1		1	

† Harnau's paper [H6] missed one design. It is not known whether the missing design or its complement is decomposable.

Table 2. Designs without repeated blocks, $10 \leq v \leq 16$			
$t-(v,k,\lambda)$	Existence	$f(v,k,t,\lambda)$	Reference
2-(10,3,2)	Yes	394	C2,I1
2-(10,4,2)	Yes	3	N2,G2,G7,I1
2-(10,5,4)	Yes	21	L2,G2,I1
3-(10,4,1)	Yes	1	B1,W2,L3
3-(10,5,3)	Yes	7	B2,G2,I1
4-(10,5,6)	Yes	1	all 5-tuples
2-(11,3,3)	Yes	?	B6
2-(11,4,6)	?		
2-(11,5,2)	Yes	1	H7,C1,G6,I1
3-(11,4,4)	?		
3-(11,5,2)	No	0	D1,G5
4-(11,5,1)	Yes	1	B1,W2,L3
2-(12,3,2)	Yes	?	B7
2-(12,4,3)	?		
2-(12,5,20)	?		
2-(12,6,5)	Yes	601	I1
3-(12,4,3)	?		
3-(12,5,6)	?		
3-(12,6,2)	Yes	1	G6,I1
4-(12,5,4)	Yes	?	D2
4-(12,6,2)	No	0	G5
5-(12,6,1)	Yes	1	B1,W2,L3
2-(13,3,1)	Yes	2	D4,B5,C3
2-(13,4,1)	Yes	1	
2-(13,5,5)	?		
2-(13,6,5)	?		
3-(13,4,2)	?		
3-(13,5,15)	?		

Table 2 (continued). Designs without repeated blocks.			
3-(13,6,10)	?		
4-(13,5,3)	Yes	?	H4
4-(13,6,6)	?		
5-(13,6,4)	?		
2-(14,3,6)	?		
2-(14,4,6)	?		
2-(14,5,20)	?		
2-(14,6,15)	?		
2-(14,7,6)	Yes	≥12	P1
3-(14,4,1)	Yes	4	M2
3-(14,5,5)	?		
3-(14,6,5)	?		
3-(14,7,5)	?		
4-(14,5,10)	Yes	1	all 5-tuples
4-(14,6,15)	?		
4-(14,7,20)	?		
5-(14,6,3)	?		
5-(14,7,6)	?		
6-(14,7,4)	?		
2-(15,3,1)	Yes	80	C4,W1,F1,H1
2-(15,4,6)	?		
2-(15,5,2)	No	0	N3
2-(15,6,5)	?		
2-(15,7,3)	Yes	5	N1,G2,I1
3-(15,4,12)	Yes	1	all quadruples
3-(15,5,6)	?		
3-(15,6,20)	?		
3-(15,7,15)	?		
4-(15,5,1)	No	0	M2

Table 2 (continued). Designs without repeated blocks.			
4-(15,6,5)	?		
4-(15,7,5)	?		
5-(15,6,10)	Yes	1	all 6-tuples
5-(15,7,15)	?		
6-(15,7,3)	?		
2-(16,3,2)	Yes	?	B7
2-(16,4,1)	Yes	1	W2
2-(16,5,4)	?		
2-(16,6,1)	No	0	$b<v$
2-(16,7,14)	?		
2-(16,8,7)	Yes	≥30	P1
3-(16,4,1)	Yes	≥31301	L1
3-(16,5,6)	?		
3-(16,6,2)	?		
3-(16,7,15)	?		
3-(16,8,3)	Yes	5	I1
4-(16,5,12)	Yes	1	all 5-tuples
4-(16,6,6)	?		
4-(16,7,20)	?		
4-(16,8,15)	?		
5-(16,6,1)	No	0	M2
5-(16,7,5)	?		
5-(16,8,5)	?		
6-(16,7,10)	Yes	1	all 7-tuples
6-(16,8,15)	?		
7-(16,8,3)	?		
2-(16,6,2)	Yes	3	H7,G2,I1
2-(16,6,3)	Yes	?	B3

Table 3. Designs with repeated blocks				
$t-(v,k,\lambda)$	$g(v,k,t,\lambda)$	Reference	$g^*(v,k,t,\lambda)$	Reference
2-(6,3,2)	0	N2 et al.	0	
2-(6,3,4)	3	G10	0	B8
2-(6,3,6)	6	G10	0	B8
2-(6,3,8)	13	I1	0	B8
2-(6,3,10)	19	I1	0	B8
2-(6,3,12)	34	I2	0	B8
2-(6,3,14)	49	I2	0	B8
2-(6,3,2q),$q\geq 8$	?		0	B8
2-(7,3,2)	3	N1,G10,I1	0	G10
2-(7,3,3)	9	M4,G10,I1	0	G10
2-(7,3,4)	34	G10	0	G10
2-(7,3,5)	107	I1	?	
2-(7,3,6)	417	I1	?	
2-(8,3,6)	$\geq$100	G10	$\geq$100	G10
2-(8,4,3)	0	N1,G10,I1	0	G10
2-(8,4,6)	2060	G13	?	
3-(8,4,2)	3	M3,G8,I1	0	G8
3-(8,4,3)	9	M3,G8,I1	0	G8
3-(8,4,4)	30	I2	0	G8
3-(8,4,5)	$\leq$107	G8	?	
3-(8,4,6)	$\leq$417	G8	?	
2-(9,3,2)	23	M1,M4,I1	16	M1,M4
2-(9,4,3)	0	G10,I1	0	
3-(9,4,6)	$\geq$50	G10	$\geq$50	G10
2-(10,3,2)	566	G1,I1	566	

Furthermore, $f(11,5,2,4) + g(11,5,2,4) = 3337$ [I1].

References

[B1] J.A. Barrau, "Over de combinatorische opgave van Steiner", *Kon. Akad. Wetensch. Amst. Verslag Wis. en Natuurk. Afd.* 17 (1908) 318-326. (= "On the combinatory problem of Steiner", *Kon. Akad. Wetensch. Amst. Proc. Sect. Sci.* 11 (1908) 352-360).

[B2] V.N. Bhat, "Nonisomorphic solutions of some BIBDs III", *Journal of Combinatorial Theory* A12 (1972) 225-252. MR 45 - 6644. Z 245.05008.

[B3] K.N. Bhattacharya, "A new balanced incomplete block design" *Science and Culture* 9 (1944) 508. MR 6 - 163. Z 60, 313.

[B4] D.R. Breach, "The 2-(9,4,3) and 3-(10,5,3) designs", *Journal of Combinatorial Theory* A27 (1979) 50-63. MR 80i:05017. Z 418.05013.

[B5] G. Brunel, "Sur les deux systèmes de triades de treize elements", *J. Math. Pures Appl.* (5) 7 (1901) 305-330. *(Mem. Soc. Sci. Phys. Nat. Bordeaux* (6) 2 (1902) 1-23).

[B6] J. van Buggenhaut, "Existence and construction of 2-designs $S_3(2,3,v)$ without repeated blocks", *Journal of Geometry* 4 (1974) 1-10. MR 48 - 8260. Z 271.05006.

[B7] J. van Buggenhaut, "On the existence of 2-designs $S_2(2,3,v)$ without repeated blocks", *Discrete Mathematics* 8 (1974) 105-109. MR 48 - 8259. Z 276.05021.

[B8] G. Burosch, "Über die Anzahl elementarer BUB der vollständingen (6,3)-Familie", *Rostock. Math. Kolloq.* 11 (1979) 5-11. MR 81c:05013. Z 431.62049.

[C1] P.J. Cameron, "Biplanes", *Math. Z.* 131 (1973) 85-101. MR 50 -12757. Z 243.05021.

[C2] C.J. Colbourn, M.J. Colbourn, J.J. Harms, and A. Rosa, "A complete census of (10,3,2) block designs and of Mendelsohn triple systems of order ten. III. (10,3,2) block designs without repeated blocks", *Congressus Numerantium* 37 (1983) 211-234.

[C3] F.N. Cole, "The triad systems of thirteen letters", *Transactions of the American Mathematical Society* 14 (1913) 1-5.

[C4] F.N. Cole, L.D. Cummings, and H.S. White, "The complete enumeration of triad systems in 15 elements" *Proc. Nat. Acad. Sci. U.S.A.* 3 (1917) 197-199.

[D1] M. Dehon, "Nonexistence d'un 3-design de parametres $\lambda=2$, $k=5$, et $v=11$", *Discrete Mathematics* 15 (1976) 23-25. MR 53 - 2705. Z 319.05011.

[D2] R.H.F. Denniston, "A small 4-design", *Annals of Discrete Mathematics* 18 (1983) 291-294.

[D3] R.H.F. Denniston, "Enumeration of symmetric designs (25,9,3)", *Annals of Discrete Mathematics* 15 (1982) 111-127. Z 488.05011.

[D4] V. de Pasquale, "Sui sistemi ternari di 13 elementi", *Rend. R. Inst. Lombardo Sci. e Lett.* (2) 32 (1899) 213-221.

[D5] J. Doyen and A. Rosa, "An updated bibliography and survey of Steiner systems", *Annals of Discrete Mathematics* 7 (1980) 317-349. Z 422.51006.

[F1] R.A. Fisher, "An examination of the different possible solutions of a problem in incomplete blocks", *Annals of Eugenics* 10 (1940) 52-75. MR 1, 348.

[G1] B. Ganter, A. Gülzow, R. Mathon, and A. Rosa, "A complete census of (10,3,2) block designs and of Mendelsohn triple systems of order ten. IV. (10,3,2) block designs with repeated blocks", Mathematische Schriften Kassel 5/78, Gesamthochschule Kassel, 1978.

[G2] P.B. Gibbons, "Computing techniques for the construction and analysis of block designs" Ph.D. thesis, Department of Computer Science, University of Toronto, 1976. (also Technical Report 92, Computer Science, University of Toronto, May 1976).

[G3] P.B. Gibbons, R.A. Mathon, and D.G. Corneil, "Computing techniques for the construction and analysis of block designs", *Utilitas Mathematica* 11 (1977) 161-192. MR 58 - 10491. Z 352.05009.

[G4] H.-D.O.F. Gronau, "Einige Bemerkungen zur Arbeit 'Über die Anzahl elementarer BUB in eingeschränkten (v,k)-Familien' von D. Rasch und G. Herrendörfer", *Rostock. Math. Kolloq.* 9 (1978) 27-34. MR 80k:05016a. Z 418.62056.

[G5] H.-D.O.F. Gronau, "Nonexistence of 3-(11,5,2) and 4-(12,6,2) designs", *Rostock. Math. Kolloq.* 18 (1981) 97-98. Z 481.05013.

[G6] H.-D.O.F. Gronau, "The 2-(11,5,2) and 3-(12,6,2) designs", *Rostock. Math. Kolloq.* 15 (1980) 77-80. MR 82j:05040a. Z 453.62060.

[G7] H.-D.O.F. Gronau, "The 2-(10,4,2) designs", *Rostock. Math. Kolloq.* 16 (1981) 5-10. MR 82j:05040b. Z 453.62061.

[G8] H.-D.O.F. Gronau, "Über (2p-1)-(4p,2p,λ)-Blockpläne", *Rostock. Math. Kolloq.* 11 (1979) 67-74. MR 81e:05034. Z 455.62063.

[G9] H.-D.O.F. Gronau, "Über nichtisomorphe elementare blockwiederholungsfreie 2-(8,4,λ)-Blockpläne I", *Rostock. Math. Kolloq.* 11 (1979) 59-66. MR 81e:05033. Z 456.05014.

[G10] H.-D.O.F. Gronau and J. Prestin, "Some results on designs with repeated blocks", *Rostock. Math. Kolloq.* 21 (1982) 15-38.

[G11] H.-D.O.F. Gronau and R. Reimer, "Über nichtisomorphe elementare blockwiederholungsfreie 2-(8,4,λ)-Blockpläne II", *Rostock. Math. Kolloq.* 17 (1981) 27-35. MR 83b:05017.

[G12]
H.-D.O.F. Gronau and R. Reimer, "Über nichtisomorphe elementare blockwiederholungsfreie 2-(8,4,λ)-Blockpläne III", *Rostock. Math. Kolloq.* 17 (1981) 37-47. MR 83b:05018.

[G13]
H.-D.O.F. Gronau and R. Reimer, "On 2-(8,4,λ) designs with repeated blocks", in preparation.

[H1] M. Hall Jr. and J.D. Swift, "Determination of Steiner triple systems of order 15", *Math. Tables Aids Comput.* 9 (1955) 146-152. MR 18, 192.

[H2] H. Hanani, "Balanced incomplete block designs and related designs", *Discrete Mathematics* 11 (1975) 255-369.

[H3] H. Hanani, "The existence and construction of balanced incomplete block designs", *Ann. Math. Stat.* 32 (1961) 361-386. MR 29 - 4161. Z 107, 361.

[H4] H. Hanani, A. Hartman, and E.S. Kramer, "On three-designs of small order", *Discrete Mathematics* 45 (1983) 75-97.

[H5] W. Harnau, "Die Anzahl paarweise nichtisomorpher, elementarer, wiederholungsfreier 2-(9,3,λ) Blockpläne, Teil I", *Rostock. Math. Kolloq.* 11 (1979) 75-83. MR 81e:05036. Z 464.62073.

[H6] W. Harnau, "Die Anzahl paarweise nichtisomorpher, elementarer, wiederholungsfreier 2-(9,3,λ)-Blockpläne, Teil II", *Rostock. Math. Kolloq.* 13 (1980) 43-47. MR 82d:05025. Z 475.62058.

[H7] Q.M. Husain, "On the totality of solutions for the symmetrical incomplete block designs: λ=2, k = 5 or 6", *Sankhya* 7 (1945) 204-208. MR 7, 233. Z 60, 314.

[I1] A.V. Ivanov, "Konstruktivnoje perečislenie sistem incidentnosti III", *Rostock. Math. Kolloq.* 24 (1983) 43-62. (also "Nekotoryje rezultaty konstruktivnogo perečislenia kombinatornykh blok-schem", thesis, Moskovskij fisiko-techniceskij institut Moskva, 1981).

[I2] A.V. Ivanov, personal communication.

[K1] E.S. Kramer, "Indecomposable triple systems" *Discrete Mathematics* 8 (1974) 173-180. MR 48 - 10863. Z 276.05020.

[L1] C.C. Lindner and A. Rosa, "Steiner quadruple systems - a survey", *Discrete Mathematics* 22 (1978) 147-181.

[L2] J.H. van Lint, H.C.A. van Tilborg, and J.R. Wiekema, "Block designs with v=10, k=5, and λ=4", *Journal of Combinatorial Theory* A23 (1977) 105-115. MR 55 - 12540. Z 359.05010.

[L3] H. Lüneburg, "Über die Gruppen von Mathieu", *Journal of Algebra* 19 (1968) 194-210. MR 38 - 226. Z 169, 35.

[M1] R.A. Mathon and A. Rosa, "A census of Mendelsohn triple systems of order nine", *Ars Combinatoria* 4 (1977) 309-315. MR 57 - 2933. Z 442.05006.

[M2] N.S. Mendelsohn and S.H.Y. Hung, "On the Steiner systems S(3,4,14) and S(4,5,15)", *Utilitas Mathematica* 1 (1972) 5-95. MR 46 - 1618. Z 258.05017.

[M3] E.J. Morgan, "Isomorphism classes of some small block designs", *Ars Combinatoria* 4 (1977) 25-35. MR 58 - 10494. Z 395.05009.

[M4] E.J. Morgan, "Some small quasi-multiple designs", *Ars Combinatoria* 3 (1977) 233-250. MR 56 - 15456. Z 394.05005.

[N1] H.K. Nandi, "A further note on nonisomorphic solutions of incomplete block designs", *Sankhya* 7 (1946) 313-316. MR 8, 127. Z 60, 314.

[N2] H.K. Nandi, "Enumeration of nonisomorphic solutions of balanced incomplete block designs", *Sankhya* 7 (1946) 305-312. MR 8, 127. Z 60, 314.

[N3] H.K. Nandi, "On the relation between certain types of configurations", *Bull. Calcutta Math. Soc.* 37 (1945) 92-94. MR 7, 357. Z 60, 313.

[P1] D.A. Preece, "Incomplete block designs with v=2k", *Sankhya* A29 (1967) 305-316. MR 37 - 2618. Z 178, 220.

[R1] D. Rasch and G. Herrendörfer, "Über die Anzahl elementarer BUB in eingeschränkten (v,k)-Familien", *Rostock. Math. Kolloq.* 8 (1978) 71-82. MR 58 - 16340. Z 394.62060.

[R2] D. Rasch and G. Herrendörfer, "Weitere Bemerkungen zur Anzahl elementarer BUB in (v,k)-Familien", *Rostock. Math. Kolloq.* 9 (1978) 35-42. MR 80k:05016b. Z 418.62055.

[W1] H.S. White, F.N. Cole, and L.D. Cummings, "Complete classification of triad systems on fifteen elements", *Memoirs Nat. Acad. Sci. U.S.A.* 14 (1919) 1-89.

[W2] E. Witt, "Über Steinersche Systeme", *Abh. Math. Sem. Univ. Hamburg* 12 (1938) 265-275. Z 19, 251.

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 221–226

Directing Cyclic Triple Systems

Janelle J. Harms

Department of Computational Science
University of Saskatchewan
Saskatoon, Saskatchewan
CANADA

Abstract

An efficient algorithm is presented for producing a directed cyclic triple system from an undirected one, in the case when $\lambda=1$. The algorithm has a worst case timing of $O(b^3)$, where b is the number of triples in the cyclic triple system.

1. Introduction

Standard definitions in design theory are employed (see [2,4] for example). A *balanced incomplete block design* with v elements, block size k, and balance factor λ is denoted by $B[k,\lambda;v]$. A *cyclic block design* is a $B[k,\lambda;v]$ with elements $\{0,1,...,v\text{-}1\}$ for which, if $\{a_1, a_2,...,a_k\}$ is a block, $\{a_1 + 1,a_2 + 1,...,a_k + 1\}$ is also a block (addition performed modulo v).

For a block $b = \{a_1,...,a_k\}$ define the set $CL(b) = \{\{a_1 + i,...,a_k + i\} \mid 0 \leq i < v$, addition modulo $v\}$. A *collection of starter blocks* for a $CB[k,\lambda;v]$ with the multiset of blocks B is a minimal multiset $S \subseteq B$ for which the multiset $\{b \mid b \in CL(s), s \in S\} = B$.

Now restrict attention to $CB[3,\lambda;v]$. Each block b has $|CL(b)| = v/3$ or v. In the former case, the block is called *short*, and belongs to $CL(\{0,v/3,2v/3\})$. Finding a $CB[3,\lambda;v]$ is equivalent to finding a suitable collection of starter blocks. Alternatively, one can represent the collection of starter blocks as a *collection of difference triples*, $DT[3,\lambda;v]$. Difference triples are derived from the starter blocks as follows. Each starter block, $s = \{a,b,c\}$ is represented by the collection of 6 differences $\{a\text{-}b,b\text{-}a,c\text{-}b,b\text{-}c,c\text{-}a,a\text{-}c\}$. To represent this set, it suffices to retain only the difference triple for this starter block which is the multiset $\{(\min(a\text{-}b,b\text{-}a)), (\min(c\text{-}b,b\text{-}c)), (\min(c\text{-}a,a\text{-}c))\}$, arithmetic modulo v. Let $\{x,y,z\}$ be a difference triple. It is evident that either $x + y + z \equiv 0 \pmod v$ or $x + y \equiv z \pmod v$, and if there are n difference triples in the system and $\lambda=2$ then $\bigcup_{i=1}^{n} \{x_i,y_i,z_i\} = \{1,1,2,2,...,(v\text{-}1)/2,(v\text{-}1)/2\}$ if v is odd or $\{1,1,...,(v\text{-}1)/2,(v\text{-}1)/2, v/2\}$ if v is even.

A *collection of directed difference triples*, DDT[3,1;v] is derived from a DT[3,2;v] corresponding to starter blocks of a CB[3,2;v] by "directing" the difference triples $\{x_i,y_i,z_i\}$ so that each difference occurs only once and $\bigcup_{i=1}^{n} (d_i,e_i,f_i) = \{1,2,...,v\text{-}1\}$, where $d_i + e_i + f_i \neq 0 \pmod v$ and $d_i = x_i$ or $v - x_i$, $e_i = y_i$ or $v - y_i$ and $f_i = z_i$ or $v - z_i$. This can be generalized to other values of λ.

A *directed triple system*, DB[3,λ;v], is analogous to a triple system B[3,λ;v], but the blocks are "directed". A directed triple of a DB[3,λ;v], (a,b,c), contains the ordered pairs (a,b), (b,c) and (a,c); each ordered pair of elements is contained in precisely λ of the blocks. A *directed cyclic triple system*, DCB[3,λ;v] is analogous to a cyclic triple system but with directed blocks.

Existence of cyclic directed triple systems have been investigated in [3]; it is shown there that a directed cyclic triple system exists whenever $v \equiv 1,4,7 \pmod{12}$. A stronger result is given in this paper: all CB[3,2;v] designs with $v \equiv 1,4,7 \pmod{12}$ can be directed into DCB[3,1;v] designs.

2. Finding Cyclic Directed Block Designs

Theorem 1: Every DT[3,2;v] underlies a DDT[3,1;v] if and only if $v \equiv 1,4,7 \pmod{12}$.

Proof:

It is known that a CB[3,2;v] exists (and thus a DT[3,2;v] exists) only when $v \equiv 0,1,3,4,7,9 \pmod{12}$ [2]. Short blocks cannot be directed and these occur when $v \equiv 0,3,9 \pmod{12}$. Thus it is necessary that $v \equiv 1,4,7 \pmod{12}$.

Given a set of difference triples DT[3,2;v], denoted T, a DDT[3,1;v] D, will be formed using a technique which we call "conflict resolution". The algorithm directs each triple of T in turn and puts them in D, resolving conflicts that arise without introducing new conflicts. To direct a triple, form a new triple t = (d,e,f) in D so that it contains no differences seen in D already and $d + e + f \neq 0 \pmod v$ holds. If this is not possible a conflict arises.

The method begins by choosing the first triple to direct, as follows. If v is odd, choose any triple t = (a,b,c). Two cases arise.

1. a,b,c are distinct. If $a + b + c \neq 0 \pmod v$, include t in D. Otherwise include (v-a,b,c).
2. a,b,c are not distinct, say a=b. Include the triple (a,v-a,c) in D. Note that $a + v - a + c \neq 0 \pmod v$.

If v is even, choose the triple t = (a,b,c) containing the difference $v/2$. Two cases arise again.

1. a,b,c are distinct. If $a + b + c \neq 0 \pmod{v}$ include t in D. Otherwise if a $= v/2$ include (a,v-b,c) otherwise include (v-a,b,c) in D.
2. a,b,c are not distinct. This case is handled like the odd case 2 with c = $v/2$.

Each remaining triple is processed in turn; the next triple to be processed is chosen to be one which has a difference e for which one of e or v-e has been previously used. If no such triple exists, any remaining triple in T may be chosen. Three cases arise.

1. No triple is found, the system is directed.
2. Two instances of a difference occur in the triple. Suppose that the triple is (a,a,b). Include (a,v-a,b) in D if the difference b has not been seen before otherwise include (a,v-a,v-b).
3. The triple found is t = (a,b,c) where a,b,c are distinct (mod v), then u = (d,e,f) is placed in D where:

   d = v - a if a has been used previously
   a otherwise
   e = v - b if b has been used previously
   b otherwise
   f = v - c if c has been used previously
   c otherwise.

   If $d + e + f \neq 0 \pmod{v}$, this directed difference triple is valid. Otherwise, $d + e + f \equiv 0 \pmod{v}$ and this triple is not valid. To resolve this conflict in D, u is fixed by replacing d by v - d in D. Let m = v - d. The following is repeated until the conflict is resolved. Find the other difference triple, w, in D containing the difference m. If there is no such triple found in D, the conflicts are resolved, since m is in an remaining triple of T which will be handled later. Otherwise, w = (g,h,i); we may assume that g = m. Then w becomes (v-g,h,i) and two cases arise.

   1. $v\text{-g} + h + i \neq 0 \pmod{v}$. The triple is valid and conflicts are resolved.
   2. $v\text{-g} + h + i \equiv 0 \pmod{v}$. w becomes ($v$-g,x,y) where x = v-h, y = i and m = x if $h \neq v/2$; otherwise x = h, y = v-i, and m = y. Continue conflict resolving with this new m.

The conflict resolving portion of the algorithm will always finish. Each triple of D is seen at most twice (in fact, only one triple of D will be seen twice). If the next triple, w = (g,h,i), has already been seen in the conflict resolution, then it will be directed without causing a conflict. If w=u, the first triple of the conflict resolution, then g = v-d and, if m=h, (g,v-h,i) is a valid triple since we know that $g + h + i \neq 0 \pmod{v}$ and $v\text{-}g + h + i \equiv 0 \pmod{v}$. The case of $m = i$ is handled the same. If $w \neq u$, two differences, say g and h, have been changed before in the conflict resolution, and $(g,h,v\text{-}i)$ is a valid triple since $g + h + i \neq 0 \pmod{v}$ and $g + v\text{-}h + i \equiv 0 \pmod{v}$ or $v\text{-}g + h + i \equiv 0 \pmod{v}$. In either case, no new conflict is introduced and the conflict resolution ends.

This method of conflict resolution never adds new conflicts, and thus the method will finish with a DDT[3,1;v]. ●

Theorem 2: Every DDT[3,1;v] can be translated into a DCB[3,1;v].

Proof:

Each block in a DDT[3,1;v], D can be ordered so that the third difference is the sum of the first two (mod v). Then for each triple (a, b, a+b) ∈ D let S contain the block (0, a, a+b). S is a set of starter blocks for a DCB[3, 1; v]. ●

The algorithm is efficient with a worst case timing of $O(b^3)$, where b is the number of triples in the DCB[3,λ;v). There are $O(b)$ triples to be directed. Each triple may conflict; resolving the conflict takes $O(b)$ time. Finally, it takes $O(b)$ time to find the next triple to direct. Thus, in the worst case, the algorithm takes $O(b^3)$ time. It is likely that this running time could easily be improved by clever implementation.

3. An Example

The following is an example of the method described in the proofs above. A CB[3,2;16] design is directed into a DCB[3,1;16] design.

Consider the starter blocks for the design:

{0,1,3} {0,1,5} {0,2,8} {0,3,10} {0,4,9}

and the corresponding difference triples are:

(1,2,3) (1,4,5) (2,6,8) (3,7,6) (4,5,7)

Since v is even the first triple is (2,6,8) which is invalid and thus becomes (14,6,8). The following triples are the results of processing according to the method until a conflict occurs in the last one:

(14,6,8) (1,2,3) (13,7,10) (15,4,5) (12,11,9)

(12,11,9) is a conflict. The sequence of changes to triples to resolve the conflict is as follows:

(12,11,9) becomes (4,11,9)

(15,4,5) becomes (15,12,5) becomes (1,12,5)

(1,2,3) becomes (15,2,3) , a valid triple

Thus the final difference triples are:

(15,2,3) (1,12,5) (14,6,8) (13,7,10) (4,11,9)

These difference triples can be reordered so that in every triple the first two differences sum to the third. After the reordering the difference triples become:

(15,3,2) (12,5,1) (14,8,6) (13,10,7) (11,9,4)

The corresponding starter blocks of this directed cyclic triple system are:

(0,15,2) (0,12,1) (0,14,6) (0,13,7) (0,11,4)

These blocks in turn can be expanded to form the directed triple system.

4. Conclusions

Colbourn and Harms [1,5] have previously shown that every triple system with even λ underlies a directed triple system with balance factor $\lambda/2$. This provides a powerful technique for translating results on undirected triple systems into results on directed triple systems. However, that technique may alter the automorphism group significantly. The algorithm embodied in theorem 1 demonstrates that directing triple systems can still be done, preserving a cyclic automorphism. Once again, this gives a powerful vehicle for applying the many results on cyclic undirected systems [4] to cyclic directed systems.

Acknowledgements

I would like to thank Charles J. Colbourn for suggesting the problem and for giving me valuable assistance in writing this paper. I would also like to thank the referees for their helpful comments.

References

[1] C.J. Colbourn and J.J. Harms, "Directing triple systems", *Ars Combinatoria* 15 (1983) 261-266.

[2] M.J. Colbourn and C.J. Colbourn, "Cyclic block designs with block size 3", *European Journal of Combinatorics* 2 (1981) 21-26.

[3] M.J. Colbourn and C.J. Colbourn, "The analysis of directed triple systems by refinement", *Annals of Discrete Mathematics* 15 (1982) 97-103.

[4] M.J. Colbourn and R.A. Mathon, "On cyclic Steiner 2-designs", *Annals of Discrete Mathematics* 7 (1980) 215-253.

[5] J.J. Harms and C.J. Colbourn, "An optimal algorithm for directing triple systems using eulerian circuits", *Annals of Discrete Mathematics*, to appear.

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 227–246
Elsevier Science Publishers B.V. (North-Holland)

Constructive Enumeration of Incidence Systems

A.V. Ivanov

Institute for System Studies
Academy of Sciences of the USSR
20 Ryleev St., Moscow 119034
U.S.S.R.

Abstract

This article presents some propositions (whose proofs are described in the author's thesis [1] and a related paper [2]) on which the constructive enumeration algorithm for incidence systems is based. A computer implementation allowed us to get some new results on the enumeration of balanced incomplete block designs.

1. Definitions and Notations. The General Scheme of Constructive Enumeration.

Let V = $\{e_1, ..., e_v\}$ be a finite set of elements, and let B = $\{B_1, ..., B_b\}$ be a collection of its (not necessarily distinct) subsets called *blocks*. A pair D = (V,B) is called a *block design* or *incidence system*.

As a rule an incidence system is characterized by the collections of numbers K = $\{k_i, 1 \le i \le b\}$, R = $\{r_j, 1 \le j \le v\}$, and $\Lambda = \{\lambda_{ij}, 1 \le i,j \le v\}$ where k_i is the cardinality of block i, r_j is the number of occurrences of element j in blocks, and λ_{ij} is the number of simultaneous occurrences of i and j in blocks. By combining conditions imposed on the parameters (numbers) from the collections K, R, and Λ, one can specify some subclasses of the class of designs. *Proper designs* have $k_i = k$ for all $1 \le i \le b$. *Regular designs* have $r_j = r$ for all $1 \le j \le v$. *Balanced (proper pairwise balanced) designs* have $\lambda_{ij} = \lambda$ for all $1 \le i,j \le v$. Supplementing these conditions with others, we get partially balanced designs, group divisible designs, t-designs, and so on.

If the set B of blocks of a balanced block design D does not contain all k-subsets of V, the design D is called *incomplete*, and is a BIB design or BIBD. The five numbers (v, b, r, k, λ) are the parameters of the BIBD, but only three of them are independent [3,4].

A classical problem of combinatorial theory is the constructive enumeration problem which consists of creating a complete list of distinct (inequivalent, nonisomorphic) combinatorial objects from a given class. Both the practical value of these lists and the difficulty of solving the constructive enumeration problems are generally known. In [5], a universal and effective approach to solving these problems for different combinatorial objects is described. The algorithm presented in this paper is a development of this approach for the enumeration of incidence systems.

Let $D = \{d_1, ..., d_l\}$ be a set of all pairwise different incidence systems with v elements and b blocks. On the sets of elements and blocks of each system acts the symmetric permutation group S_v and S_b, respectively. Their action induces the group $G_D = S_v \times S_b$ which acts on the set D. Two incidence systems of D belonging to the same orbit of the group G_D are said to be *equivalent* (isomorphic): $d_i \sim d_j$ if and only if there exists a $g \in G_D$ for which $d_i = gd_j$. An element g of the group is an *automorphism* of d_i if $d_i = gd_i$. The set of all automorphisms of the incidence system d_i forms a group.

In order to consider restricted classes of incidence systems, such as BIB designs, we shall introduce a membership predicate P defined on elements of D which is invariant under the action of the group G_D. The domain of D for which P is true, $D_P = \{d_i \mid P(d_i)\}$, determines some class of incidence systems. Thus the constructive enumeration problem of block designs belonging to the class D_P consists of finding an arbitrary transversal of the orbits of the group G_D on D_P.

We shall introduce a canonicity predicate C defined on the elements of D which is true for a single *(canonical)* element of every equivalence class. Obviously, $D_P \cap D_C = D_Q$ (where $Q = P\&C$) is a transversal of the orbits of the group G_D on D_P. We shall seek a transversal of the kind described.

Let the set D be partitioned into disjoint subsets $\{D_i\}$, each of which contains all block designs of D with the same occurrence of the first element in blocks. In the second step, every D_i is to be partitioned into disjoint subsets in accordance with the occurrence of the second element in the blocks, and so on. The system U of these subsets of the set D which is obtained in step v of such a partitioning is ordered by inclusion in the form of a tree with D as root, and all one-element subsets $\{d_i\}$ serve as nodes of degree one.

We introduce a set $R = \{R_i\}$ of predicates defined on the nodes of this tree U so that for all $D^* \subset U$, and for all $R_i \in R$, if there is a $d \in D^*$ for which $Q(d)$ holds, then $R_i(D^*)$ holds. The predicate $R_i \in R$ is called an *extension* of predicate Q on U. From the definition of the set R, it is clear that each extension of the predicate Q provides some necessary conditions for the existence in D^* of at least one element on which Q is true. Conversely, falsity of some predicate of R on some node D^* of U serves as a sufficient condition for Q to be false on all elements of subset D^*.

The constructive enumeration problem solving procedure consists of a backtracking search through the tree U applying to every searched node one or more predicates from R. If in a current node D^* an applied predicate R_i is false, the subtree with root D^* is not searched (a regular search reduction). An examination of a node of degree one consists of applying predicate $Q = P\&C$ to an appropriate element of D.

A description of the set D by the list of its elements is unrealizable in practice. In the next paragraphs of the article, we shall describe an algorithm $A=A(D,U)$ that constructs the elements of the set D according to a correspondence with the tree U. This correspondence is that all elements of a subset $D^* \subset U$ are constructed successively one after another.

2. Canonical representation of incidence systems. Extensions of predicates.

Using the generally accepted representation of incidence systems by (0,1)-matrices, we shall introduce the canonicity predicate C in the following way. Let $N(A)$ be the number whose binary representation is obtained by reading the matrix A line by line. Then on the set D the order of elements is defined naturally:

$$d_{i_1}<d_{i_2}< \cdots <d_{i_l} \quad iff \quad N(M_{i_1})<N(M_{i_2})< \cdots <N(M_{i_l}),$$

where M_{i_j} is the incidence matrix of the system d_{i_j}.

The incidence system d_i (matrix M_i) is said to be *canonical* if it is maximal in its orbit induced by the group G_D on D: $d_j < d_i$ for all $d_j \in D$ satisfying $d_j \sim d_i$ and $i \neq j$. The canonicity predicate C is introduced by $C(d_i)$ is true exactly when for all $g \in G_D$, $gd_i \leq d_i$.

Let us describe the properties of canonical incidence matrices by the next propositions. Here and in what follows we shall denote the (0,1)-matrix of dimensions $v \times b$ by A.

Proposition 2.1: If $C(A)$ holds, then for all $w<v$, $C(A^w)$ holds, where A^w is the $w \times b$ matrix consisting of the first w rows of the matrix A.

Let a_i denote the ith row of A, and let a_j^T denote the jth column. On the set of rows and the set of columns, we interpret the order "$\leq$" lexicographically.

Proposition 2.2: If $C(A)$ holds, then for all i,i' if $i<i'$, $a_i \geq a_{i'}$. Similarly, if $C(A)$ holds, then for all j,j', if $j<j'$, $a_j^T \geq a_{j'}^T$.

The columns i and j of the matrix A are called *w-equivalent,* if the columns i and j of the matrix A^w are equal. The induced equivalence partitions the set of column numbers into classes $N^w = (Y_s^w)$. A connection between the partitions N^w and N^{w+1} is given by

Proposition 2.3: The predicate $C(A) =$ (for all s: $1 \leq s \leq |N^w|$ and $Y_s^w \in N^w$, $(Y_s^w = Y_{s'}^{w+1})$ or $(Y_s^w = Y_{s'}^{w+1} \cup Y_{s'+1}^{w+1})$, $Y_{s'}^{w+1}, Y_{s'+1}^{w+1} \in N^{w+1}$).

Let x_{is}^w denote, for $i > w$, the number of ones in the intersection of row i and the columns from Y_s^w. Assign to row i the vector $X_i^w = (x_{is}^w)$, $1 \leq s \leq |N^w|$.

Proposition 2.4: If $C(A)$, then for all $w < v$ the vector X_{w+1}^w defines the row $w+1$ of the matrix A in a unique way.

Proposition 2.5: If $C(A)$ then for all w,i with $i > w+1$, $X_i^w \leq X_{w+1}^w$.

We proceed to describe the set of most widely used extensions of the predicates for membership P and for canonicity C. Different specifications of this set allow us to enumerate the incidence systems of different types. It is readily seen that the dimensions, the occurrences of elements in blocks, the cardinalities of blocks, the occurrences of pairs of elements in blocks are general properties of block designs, and the predicates D_v^b, R, K_v and Λ corresponding to these properties are said to be *basic*. The basic predicates are extensions of the membership predicate P. Now, for every particular type of incidence system, predicate P may be written as

$$P = D_v^b \& R \& K_v \& \Lambda \& S$$

where S is a predicate describing some additional properties.

Let $A = (a_{ij})$ be a (0,1)-matrix of dimensions $v_1 \times b_1$. The set of basic predicates defined on the (0,1)-matrices consists of

a) a dimension predicate D_v^b which is true only on matrices with b columns and not more than v rows ($v_1 = v$ if a construction is completed).

b) a "row weight" predicate R which is true if and only if the number r_i of ones in row i is an element of some given set $\bar{R}$.

c) a "column weight" predicate K_v given by

$$K_v(A) = \sum_{i=1}^{v_1} a_{ij} = k_j : k_j^v \epsilon \bar{K}_v, 0 \leq k_j^v - k_j \leq v - v_1$$

where k_j is the number of ones in column j and $\bar{K}_v$ is some given set.

d) a "row dot product" predicate Λ which is true if and only if the dot product λ_{ij} of different rows i and j is an element of some given set $\bar{\Lambda}$.

The truth of extension C_w of the canonicity predicate C is determined similarly to the truth of predicate C.

It is clear that the truth on A^* of some extension Q of predicate P & C does not guarantee the existence of a completion $A^* \to A$, for which $P\&C(A)$ is true. The more often the impossibility of such a completion will be determined without its execution, the more effective the algorithm of constructive enumeration will be. A number of propositions from §4-7 allow us to construct the set of completion predicates.

For example, the next situation is met frequently. It is required to add n ($n \geq 2$) rows to the (0,1)-matrix A^* with b columns so that the obtained matrix A will satisfy the predicate K_v & Λ. Here the set K_v may be such that the condition $K_v(A)$ implies that some of the columns of matrix A^* must be completed only by ones (zeroes).

Proposition 2.6: Let $\bar{\lambda}$ ($\underline{\lambda}$) be the greatest (least, respectively) element of the set Λ. The completion $A^* \to A$ satisfying $K_v\&\Lambda(A)$ is impossible if the matrix A contains at least $\bar{\lambda}+1$ ($b-\underline{\lambda}+1$, respectively) columns which are forced to be completed by ones (zeroes, respectively).

3. Construction of the search tree

We shall describe an algorithm for constructive enumeration of incidence systems for the case of BIBDs, and then we shall generalize it to the arbitrary case.

The solution of the constructive enumeration problem for BIBDs with parameters (v,b,r,k,λ) is a complete list of incidence matrices satisfying predicate P & C. We shall construct these matrices, verifying the extensions D_v^b, R, K_v and Λ of the membership predicate P. For BIBDs, $\bar{R} = r$, $\bar{K}_v = k$, and $\Lambda = \lambda$. Questions concerning the truth verification of the extension C_w will be discussed in §7. Here we assume that canonicity is verified on the completed matrices. We shall use only the necessary conditions of canonicity (propositions 2.2 - 2.5). Note that from these conditions and the definition of BIBD, it follows that the two first rows of the incidence matrix will be the following:

1 1 1 ... 1	1 1 1 ... 1	0 0 0 ... 0	0 0 0 ... 0
1 1 1 ... 1	0 0 0 ... 0	1 1 1 ... 1	0 0 0 ... 0
λ	$r-\lambda$	$r-\lambda$	$b-2r+\lambda$

Assume that w rows of incidence matrix A are already constructed, and that the column set of this matrix is partitioned into equivalence classes $N^w = (Y_s^w)$. Recall that x_{is}^w ($i > w$) denotes the number of ones in the intersection of row i and columns from Y_s^w, and to every row i, the vector $X_i^w = (x_{is}^w$, $1 \leq s \leq |N^w|$ is assigned. This vector is essentially a projection of row i on the partition N^w. It is easy to see that the dimension of this vector is $|N^w|$, and it does not depend on i.

Let X^w be the set of all possible distinct projections of the last $v-w$ rows of the incidence matrix of a BIBD. Let $N_i^w \subseteq N^w$ $(i \leq w)$ be column equivalence classes such that $a_{ij} = 1$ for all $j \in Y_s^w \in N^w$. Let χ^w denote the set of integral nonnegative solutions of the system

$$\sum_{Y_s^w \in N^w} x_s^w = r$$

$$\sum_{Y_s^w \in N_i^w} x_s^w = \lambda, \ for \ 1 \leq i \leq w \tag{3.1}$$

$$0 \leq x_s^w \leq |Y_s^w|, \ for \ 1 \leq s \leq |N^w|$$

Proposition 3.1: $X^w \in \chi^w$ for $3 \leq w \leq v$.

By proposition 2.4 the vector X_{w+1}^w uniquely determines row $w+1$ of the canonical matrix. Thus all canonical matrices of BIBDs with given parameters are found among the matrices given by a backtracking procedure, one step of which follows. Let w rows of the matrix A be already defined, and the sets N^w, N_i^w $(i \leq w)$ be constructed. By solving the system (3.1) we determine χ^w. If χ^w is empty, then the matrix A^w cannot be completed up to A. Otherwise, the lexicographically minimal solution is selected from χ^w, and using it as X_{w+1}^w we determine the row $w+1$. After having defined the sets N^{w+1}, N_i^{w+1} $(i \leq w+1)$ in accordance with proposition 2.3, we are ready to make the next step of the search. When all possible ways of completing the matrix A are exhausted, we change row $w+1$ by taking the next solution from χ^w as X_{w+1}^w.

When $w = v$, we determine the canonicity of the obtained matrix using the algorithm from [6] (an identical algorithm is contained in [7]). After all possibilities of constructing the 3rd, 4th, ..., vth row during the search are exhausted (which can be done since χ^w is finite), the constructive enumeration problem for BIBDs with parameters (v,b,r,k,λ) will be solved.

The algorithm described has some deficiencies:

a) No efficient method to solve the system (3.1) is known.

b) A large part of the calculation leads to the construction of noncanonical matrices.

Proposition 3.2 given below allows us to get the set χ^w recursively using the previously calculated set χ^{w-1}. Thus we succeed in eliminating the first noted deficiency.

A number of essential search reductions for the described algorithm is made possible by this approach to determining χ^w, in particular a substantial reduction of the second noted deficiency. Moreover, this algorithm can be generalized for the enumeration of general incidence systems.

Proposition 3.2: The set of integral solutions of the system

$$\sum_{Y_s^w \in N_s^w} x_s^w = \lambda$$

$$x_s^w + x_{s+1}^w = \bar{x}_{s'}^w \qquad \textit{for } Y_{s'}^{w-1} = Y_s^w \cup Y_{s+1}^w \tag{3.2}$$

$$\cdot$$

$$\cdot$$

$$x_p^w = \bar{x}_{p'}^{w-1} \qquad \textit{for } Y_{p'}^{w-1} = Y_p^w$$

$$0 \le x_i^w \le |Y_i^w| \qquad \textit{for } 1 \le i \le |N^w|$$

equals the set χ^w of solutions of (3.1), where in the right hand sides, all vectors $\bar{X}^{w-1} = (\bar{x}_i^{w-1}) \in \chi^{w-1}$ are substituted consecutively.

Using proposition 3.2 all sets χ^w for different values of w may be found by solving (3.2) with $\chi^0 = \{r\}$, $\chi^1 = \{{}_{r-\lambda}^{\;\lambda}\}$.

It is clear that the system (3.2) is considerably easier to solve, because the values of some variables occurring in the system are determined at once from the available equalities. Further, each equation except the first one contains at most two variables, and the "concord" of the values of the variables is made only through the first equation. Thus instead of solving the system (3.1) in order to determine χ^w, we have to solve $n = |\chi^{w-1}|$ times the system (3.2) with different right hand sides.

In the conclusion of this section, we want to note that we are familiar with Gibbons's algorithm for enumeration of BIB designs [7],[8]. He finds the same orbits of the group G_D acting on D, and the canonicity predicate C is introduced in the same way. However, his method of finding the next $w+1$'th row is an exhaustive search of binary vectors which have length b and weight r not exceeding row w, and hence is less efficient than ours.

4. Elimination of inadmissible solutions

The recursive method of constructing the sets $\chi^2, \ldots, \chi^w$ described above makes it possible to eliminate from consideration a number of ways of obtaining the last $v-w$ rows of the matrix A, and thus to shorten considerably the search and to reduce the computer memory needed for storing solutions of the system. We shall describe two methods of such elimination which are based on a modification of the system (3.2) and/or its right hand sides. An important positive feature of these methods is that some ways of constructing the last rows are eliminated without their actual determination.

Let y_s^w be the number of ones in every column of class $Y_s^w \in N^w$, that is $y_s^w = |\{i \leq w, \text{ for all } j \in Y_s^w, a_{ij} = 1\}|$.

Proposition 4.1: $k+w-v \leq y_s^w \leq k$ for all w and for all $Y_s^w \in N^w$.

If it is not excluded to complete the matrix A^w up to the canonical one for which the vector X^w will be equal to the projection of row i on the partition N^w, then solution X^w of the system (3.2) will be called *i-admissible*. Otherwise, the solution X^w will be called *i-inadmissible*. The solution X^w of the system (3.2) will be called *admissible*, if it is i-admissible for some i; otherwise it is *inadmissible*.

Corollary: A solution $X^w = (x_s^w)$ of the system (3.2) is inadmissible if the columns of some class $Y_{s_1}^w$ ($Y_{s_0}^w$) contain exactly k ($k+w-v$) ones, and $x_{s_1}^w \neq 0$ ($x_{s_0}^w \neq Y_{s_0}^w$).

Note that if the conditions of the corollary take place, then by modifying the system (3.2) slightly one can ensure tha absence of the inadmissible solutions described in this corollary. It suffices to substitute in the right hand side of (3.2) only the vectors $\bar{X}^{w-1} = (\bar{x}_s^{w-1}) \in \chi^{w-1}$ whose sth coordinate satisfies the following conditions:

a) $\bar{x}_{s'}^{w-1} = 0$ *if* $Y_{s'}^{w-1} = Y_{s_1}^w$ $(x_{s'}^{w-1} = |Y_{s'}^{w-1}|$ *if* $Y_{s'}^{w-1} = Y_{s_0}^w)$

b) $\bar{x}_{s'}^{w-1} \leq |Y_{s_1+1}^w|$ *if* $Y_{s'}^{w-1} = Y_{s'}^{w-1} \cup Y_{s_1+1}^w$

$(\bar{x}_{s'}^{w-1} \geq |Y_{s_0}^w|$ *if* $Y_{s'}^{w-1} = Y_{s_0-1}^w \cup Y_{s_0}^w)$

Moreover, in case a) equations are to be removed from the system (3.2) altogether, and in case b) to be replaced by:

$$x_{s_1}^w + x_{s_1+1}^w = \bar{x}_{s'}^{w-1} \rightarrow x_{s_1+1}^w = \bar{x}_{s'}^{w-1}$$

$$(x_{s_0-1}^w + x_{s_0}^w = \bar{x}_{s'}^{w-1} \rightarrow x_{s_0-1}^w = \bar{x}_{s'}^{w-1} - Y_{s_0}^w)$$

Elimination of the indicated unknowns from (3.2) corresponds to an exclusion of the class $Y_{s_1}^w$ ($Y_{s_0}^w$) from N^w. One must also remember to modify the first equation of (3.2):

$$\sum_{Y_s^u \in N_u^w} x_s^u = \lambda - \lambda^u \qquad \textit{where} \quad u > w, \quad \lambda^u = \sum_{i<u} \sum_{y_s^i = k+i-v} |Y_s^i|$$

From the way the search of the sets χ^w $(1 \le w \le v)$ is performed by means of (3.2), one can see that such a modification of the system is equivalent to an elimination of inadmissible solutions from χ^w. Some coordinates of retained solutions will be identical and will stay such until the end of the search with the given matrix A^w. The sets corresponding to these unknowns will not be partitioned further, and the forced completion of these matrix columns by ones or zeroes will take place. This permits us to get rid of the noninformative coordinates and thus to economize on memory and time when solving (3.2).

Let us agree that each vector from any set χ^w will contain only the informative coordinates x_s^w. In other words, each column of the set $Y_s^w \in N^w$ for any coordinate s will contain y_s^w ones, and $k+w-v<y_s^w<k$. When we speak about the projection of a vector on the partition N^w we shall also take into account the forcibly completed column sets.

The other class of inadmissible solutions is much more extensive. The inadmissibility is related to the canonicity of the matrix A. Let the row w of A^w be constructed in accordance with $X_w^{w-1} \epsilon \chi^{w-1}$.

Proposition 4.2: The vector X_*^w is inadmissible if it satisfies (3.2) with a $\tilde{X}_*^{w-1} \epsilon \chi^{w-1}$ such that $\tilde{X}_*^{w-1} > X_w^{w-1}$.

Since we no longer need the set of all solutions of (3.2), we shall henceforth denote by χ^w the set of all admissible solutions.

Propositions 3.2 and 4.2 and the method described for modification of the system (3.2) allow us to accomplish one search step in the following way. Let the matrix A^{w-1} be constructed, and let χ^{w-1} be the set of admissible solutions. Fixing one of them as X_w^{w-1}, we shall construct A^w. Having solved (3.2) with all $X^{w-1} \epsilon \chi^{w-1}$ such that $X^{w-1} \le X_w^{w-1}$ we obtain the set χ^w of admissible solutions (here, the system should be modified if necessary).

Let us sum up the technique discussed above.

We have described a constructive enumeration algorithm for BIBDs based on the examination of the search tree, the nodes of which are the ways of row construction obtained by solving the system (3.2). We succeeded in cutting off some branches of this tree by using the corollary of proposition 4.1 (proposition 4.2) having proved that they do not contain the canonical incidence matrices of the BIBDs. However, these measures are not sufficient even for an enumeration of small BIBDs that are interesting in practice. The fact is that, generally speaking, the sets of admissible solutions to systems (3.2) are too large, which renders difficult both their storage in the computer memory and the search execution.

Three methods for overcoming the difficulties noted will be given and discussed in detail in subsequent sections. The first of them (§5) is based on the proof of w-inadmissibility for many vectors from χ^{w-1}. It may be used efficiently when the number of vectors in each set χ^{w-1} does not exceed several hundred. It is possible to prove w-inadmissibility for approximately 90-95% of these vectors. Two other methods (§6) are tailored to the enumeration of BIBDs for which it is impossible to keep complete lists of admissible solutions of the systems (3.2). In enumeration of several BIBDs, one succeeds in overcoming the difficulties noted by using the extension C_w of canonicity predicate C, in particular by using the automorphism group of the matrix A^w.

5. Row-inadmissible solutions

It is easy to prove the w-inadmissibility of many vectors from χ^{w-1} by using

Proposition 5.1: The vector X_*^{w-1} is w-inadmissible if for some coordinate s and for all vectors $X^{w-1} = (x_i^{w-1}) \in \chi^{w-1}$, $X^{w-1} \leq X_*^{w-1}$, one of the following conditions holds:

a) $x_s^{w-1} = 0$

b) $x_s^{w-1} = |Y_s^{w-1}|$

A BIBD is called *symmetric* if $v=b$. It is easy to show [4] that any two blocks of a symmetric BIBD have λ elements in common. We shall use this fact to show the w-inadmissibility of several solutions of the system (3.2).

Let z_{s1}^w be the number of common ones in the columns of the classes Y_s^w and Y_1^w, that is

$$z_{s1}^w = |\{i \leq w: \text{ for all } j_1 \in Y_s^w \text{ and all } j_2 \in Y_1^w, a_{ij_1} = a_{ij_2} = 1\}|$$

Proposition 5.2: If A is the incidence matrix of a symmetric BIBD, then for any row w we have:

a) for all $Y_s^w \in N^w$, $|Y_s^w| \geq 2 \rightarrow k + w - v \leq y_s^w \leq \lambda$

b) for all Y_s^w and $Y_1^w \in N^w$, $s \neq 1 \rightarrow \lambda + \max(y_s^w, y_1^w) - k \leq z_{s1}^w \leq \lambda$

Corollary 1: The solution $X^w = (x_i^w)$ of the system (3.2) is inadmissible if every column of some class $Y_s^w \in N^w$ contains exactly λ ones (that is, $y_s^w = \lambda$), and $x_s^w \geq 2$.

Corollary 2: The solution $X^w = (x_i^w)$ of the system (3.2) is inadmissible if for some classes Y_s^w, $Y_1^w \in N^w$ $(s \neq 1)$, $z_{s1}^w = \lambda$ and both of x_s^w and x_1^w are nonzero.

Corollary 3: The solution $X^w = (x_i^w)$ of the system (3.2) is inadmissible if for some classes Y_s^w, $Y_1^w \in N^w$ $(s \neq 1)$, $z_{s1}^w = \lambda + y_s^w - k$ and exactly one of x_s^w and x_1^w is nonzero.

Proposition 5.3: While enumerating the symmetric BIBDs, the vector X_*^{w-1} is w-inadmissible if for all $X^{w-1} = (x_i^{w-1} \in \chi^{w-1}$ with $X^{w-1} \leq X_*^{w-1}$, one of the following conditions holds:

a) there exists an s for which $z_{s1}^{w-1} < \lambda$, and $x_s^{w-1} + x_1^{w-1} \neq 0$, and exactly one of x_s^{w-1}, x_1^{w-1} is zero.

b) there exists an s for which $|Y_s^{w-1}| \geq 2$ and $y_s^{w-1} < \lambda$ and $x_s^{w-1} \leq 1$.

Let us note that there is a similarity between successive uses of propositions 4.1 and 5.1 on the one hand and propositions 5.2 and 5.3 on the other. In each of these cases, some characteristic of the columns (y_s^w or z_{s1}^w) was introduced and by the first proposition of each pair the limits of its possible values were determined. The "mobile" (depending on the constructed row number) lower limit coincides with the upper one, when the kth "one" was added in some of the considered columns. When the value of this characteristic coincides with one of the limits, some solutions of the system (3.2) were excluded by the corollary (corollaries) of this proposition from further search. Use of each of these solutions for the construction of the next matrix rows would put the considered characteristic beyond the determined limits. On the other hand, when the value of the characteristic was within the determined interval, some ways of construction of row w were excluded from the search by the second proposition of each pair. Their use in the construction of this row would, because of proposition 4.2, make it impossible for the characteristic value to reach its upper limit. The latter, as is clear from the definition of the characteristic, is a necessary condition for completing the matrix A^w to an incidence matrix of a BIBD.

The noted analogy is confirmed by the following. As in the case of proposition 4.1, by a modification of the system (3.2) one may avoid solutions which are inadmissible in the sense of proposition 5.2. For this, it suffices to add to the constraints of the system (3.2) the following conditions:

a) $x_s^w \leq 1$ for the sth component of the vector described in corollary 1.

b) $x_s^w \cdot x_l^w = 0$ for the sth and lth components of the vector described in corollary 2.

c) $(x_s^w \neq 0) \rightarrow (x_l^w \neq 0)$ for the sth and lth components of the vector described in corollary 3.

6. Two modifications of the general algorithm

The weak side of the algorithm described is the necessity of storing in the computer memory the complete lists of solutions of the system (3.2), in order to obtain the sets χ^w recursively. Using these lists, we succeeded in increasing substantially the efficiency of the search, by performing it only via admissible (propositions 4.1, 4.2 and 5.2) and w-admissible (propositions 5.1 and 5.3) lines. This made possible the solution of the problem of constructive enumeration for a number of parameter sets of BIBDs. However, the enumeration of other BIBDs by this algorithm is infeasible because it is impossible to store the complete solution lists in the computer primary memory. Below, two modifications of the general algorithm are described which allow one to manage with partially constructed lists.

For the description of the first modification, the following will be required:

Proposition 6.1: The first element x_{i1}^w of the projection $X_i^w = (x_{is}^w)$ of any ith ($w < i \leq w + k - y_1^w$) row of the canonical incidence matrix of a BIBD is not equal to 0.

Proposition 6.1 allows one to modify the algorithm for enumeration as follows. We build up the system (3.2) for finding the construction modes of the following rows as we did before, denoting by x_s^w ($1 \leq s \leq |N^w|$) the number of ones which were put in the columns of nonforcibly completed sets Y_s^w. But we solve it with the additional restriction $x_1^w \geq 1$. By proposition 6.1, we shall receive all possible projections on N^w of the next $k - y_1^w$ rows of canonical matrix A. Having constructed the $(w+1)$-th row in accordance with some X_{w+1}^w, we solve a new system (3.2) with the additional restriction $x_1^{w+1} \geq 1$, substituting in the right hand side only those solutions obtained in the previous step for which $\bar{X}^w \leq X_{w+1}^w$ (proposition 4.2). We proceed in this way until the kth "one" has been placed in the leftmost column j^* from Y_1^w. When this happens at last in row i^*, we shall find the leftmost column j^{**} of the matrix A^i which does not yet have k ones (it can turn out that this column must be forcibly completed by ones, although it is not necessary that $j^{**} \in Y_1^{i^*}$). Also, we determine the row i^{**} in which the columns j^* and j^{**} were found to be for the first time in different classes. We solve again the system (3.2) for this row, adding the condition $x_1^{i^{**}} = 0$. Using the obtained solutions, we solve again successively all systems (3.2) up to row i^* inclusive. While solving them we put restrictions on coordinates so as to obtain all ways of constructing the next $k - y_1^{i^*}$ rows of the matrix A having ones in column j^{**} and in rows numbered i for $i^* < i \leq i^* + k - y_1^{i^*}$.

Having obtained in this manner all ways of constructing the row i^*+1, we fix one of them and solve the system (3.2) with the restriction $x_1^{i^*+1} \geq 1$, and so on.

Of course, the method of constructing row i^* should be changed if the set of solutions to the system (3.2) with new constraints is empty in some stage of the computation.

We shall call the modification of the general algorithm which was just described *pumping* because of the analogy between the back and forth motion of a sucker and the multiple pumping of the solutions with prescribed properties through the rows already constructed.

The use of pumping expands the range of BIBDs which can be enumerated due to the possibility of storing the intermediate results in the computer primary memory. However, the number of solutions of the system (3.2) grows rapidly with the increasing of dimensions of BIBDs. This makes us look for some other, more complicated, modifications of the general algorithm. Now we shall describe one of these modifications.

Let w rows of the incidence matrix A of a BIBD be constructed and let the set of all nonforcibly completed columns of the matrix A^w be partitioned into classes $N^w = (Y_s^w)$. Let, by solving the system (3.2), the set χ^w of admissible projections of the next $v-w$ rows on the partition N^w be found. We shall write the set χ^w in the form of a matrix L^w of dimensions $|N^w| \times |\chi^w|$. Each column of this matrix is one of the admissible projections, and all columns are lexicographically ordered. Let y_i^w denote the number of ones in each column of the class Y_i^w $(1 \leq i \leq |N^w|)$. Then $Y_i^w \cdot (k-y_i^w)$ is the total number of ones which must be added in the next $v-w$ rows in the columns of the class Y_i^w in order to obtain the matrix A. By m_j^w we shall denote the number of rows of the matrix A having a projection on N^w which agree with row j $(1 \leq j \leq |\chi^w|)$ of the matrix L^w. The ways of constructing the rows themselves will be called the *descendants* of jth construction mode of row $w+1$. By this name, we emphasize the recursive method of their determination with the help of the systems (3.2) from the jth mode.

The values of coordinates of the recently defined vector $m^w = (m_j^w)$ must satisfy the following equations:

$$\sum_{j=1}^{|\chi^w|} m_j^w = v - w \tag{6.1}$$

$$L^w \cdot m^w = d^w$$

where the vector $d^w = ((Y_i^w) \cdot (k-y_i^w))$, $1 \leq i \leq |N^w|$.

The first equation of (6.1) means that the total number of descendants used for construction of the rows must be equal to the number of incomplete rows of the matrix A. The remaining equations express the necessity of including the missing "ones" in all classes by the descendants of selected solutions (those with $m_j^w \neq 0$.

Using (6.1) one can modify the general algorithm in the following way. Having solved the system (6.1) to obtain a solution in nonnegative integers, we shall obtain a set of solutions $M^w = (m^{wi})$. Having fixed some solution m^{wi^*}, we shall construct row $w+1$ of matrix A in accordance with the jth mode defined by the condition

$$m_j^{wi^*} \neq 0, m_{j+1}^{wi^*} = 0, \quad for \ 1 \leq l \leq |\chi^w| - j$$

In the right hand side of the system (3.2) for the determination of all possible projections of the next rows on the partition N^{w+1}, we shall substitute only those construction modes with $m_l^{wi^*} \neq 0$, $1 \leq l \leq j$. On the newly obtained set χ^{w+1}, we solve system (6.1), and so on.

The solution m^{wi^*} must be changed if the system (3.2) cannot be solved, or if the system (6.1) cannot be solved on the set of solutions to (3.2).

Thus the previous search on the construction modes of the rows is replaced by a search on the solutions M^w. This modification of the general algorithm we shall call the *method of selected descendants*. Let us note that the idea to use the solutions of the system (6.1), on which the method of selected descendants is based, is a generalization of proposition 5.1. In this proposition, however, only the necessary conditions of the solvability of (6.1) were used.

7. Use of the canonicity predicate

When solving constructive enumeration problems for combinatorial objects, the selection of nonisomorphic configurations is a very complex procedure. Using the canonical representation of incidence systems, we succeeded in avoiding pairwise comparison of the constructed objects. Using proposition 5.2, we exclude from the search a large number of construction modes of different rows of the matrix that cannot belong to canonical matrices which reduces the search greatly.

The extension C_w of canonicity predicate C was introduced in §2. However, its verification is time expensive, and therefore the use of C_w demands special discussion. In [1], a number of simple heuristic methods for using the extension C_w is given. The use of automorphism groups of partially constructed systems is the most interesting of them. As far as we know, similar methods were not presented before in enumeration problems; therefore we shall discuss it in detail.

Let A^w be a canonical $w \times b$ matrix which has a nontrivial automorphism group $G^w = (G_w, H_i^w)$, and let $N^w = (Y_s^w)$ be the partition of columns into equivalence classes. Let us consider some elements from the group H_i^w, which describes the action of group G^w on the columns of the matrix A^w. These elements permute the classes (Y_s^w) as a whole without changing the order of columns within each class. It is easy to show that the set of all such elements forms the group J_w with the multiplicative operation defined by the product of permutations. The group J_w induces the group J_{N^w} acting on the classes (Y_s^w), if to all columns from Y_s^w we assign the number s $(1 \leq s \leq |N^w|)$.

Proposition 7.1: The vector $X^w \in \chi^w$ is inadmissible if there is a permutation of its coordinates $h \in J_{N^w}$ such that $hX^w \notin \chi^w$.

Proposition 7.2 is analogous to proposition 5.2.

Proposition 7.2: The vector $X^w \in \chi^w$ is inadmissible if it satisfies (3.2) with $\tilde{X}^{w-1} \in \chi^{w-1}$ such that there exists an $h \in J_{N^{w-1}}$ for which $h\tilde{X}^{w-1} > X_w^{w-1}$.

Proposition 7.3: The vector $X^{w-1} \in \chi^{w-1}$ is w-inadmissible if for some $h \in J_{N^{w-1}}$, $hX^{w-1} > X^{w-1}$.

Using propositions 7.1-7.3, one can modify the general algorithm in the following way. In the current stage of computation we obtain the set χ^w from which we exclude inadmissible vectors in accordance with proposition 7.1, using the group J_{N^w} constructed from the automorphism group G^w of matrix A^w. The remaining vectors will be partitioned into equivalence classes: $X_1^w \sim X_2^w$ if and only if there exists an $h \in J_{N^w}$ for which $X_1^w = hX_2^w$, with $X_1^w, X_2^w \in \chi^w$. The lexicographically maximal vector from each class will be called *canonical*. We order χ^w in two stages. In the first stage we order the canonical vectors lexicographically. In the second stage we include the noncanonical vectors in the obtained list in such a way that each of them will be to the left of the canonical vector that is isomorphic to it, but to the right of the preceding canonical vector. From proposition 7.3 it follows that we must construct the next row of matrix A using only the canonical vectors. Fixing one of them as X_{w+1}^w, in order to obtain χ^{w+1} we solve the system (3.2) with all $\tilde{X}^w \in \chi^w$ such that $\tilde{X}^w \leq X_{w+1}^w$ in accordance with the introduced order. Thus we shall not obtain the inadmissible solutions described in proposition 7.2.

Let us make some comments.

1. From the construction of the group J_{N^w}, it follows that the columns from the classes Y_s^w, Y_1^w contain the same number of ones if these classes are in the same orbit. We excluded from the search the "non-informative" coordinates (by proposition 4.1 and its corollary). It means that one must omit the corresponding coordinates in each permutation $h \in J_{N^w}$ using the elimination of solutions with the help of the automorphism group.

2. The elimination of the solutions by the automorphism group of the partially constructed BIBD may be performed even when using "pumping". But instead of the group J_{N^*}, its subgroup $J^1_{N^*}$ should be considered. Each element of $J^1_{N^*}$ fixes the senior nonforcibly completed class.

3. Using the method of selected descendants and the elimination of solutions by the automorphism group simultaneously is impossible. The absence of the vector hX^w in the set χ^w does not allow one to affirm the inadmissibility of vector X^w as it was made in proposition 7.1. Some vector $X_j^{i^*-1} < X_{i^*}^{i^*-1}$ for which $m_j^{i^*-1} = 0$ could be the ancestor of the vector hX^w.

8. The results of the constructive enumeration of BIBDs

The algorithm for constructive enumeration described in this article was programmed in assembly language on an ICL 4/70 (capable of executing 300,000 operations per second) and was used for compiling complete lists of BIBDs with certain parameter sets. Information about the families of BIBDs which were enumerated is presented in Table 1.

In the column $|S|$ of this table the number of pairwise nonisomorphic designs with the given parameters is presented. The number of nonisomorphic designs without repeated blocks is given in parentheses in the same column whenever some of the solutions have repeated blocks. The processor time used is given in column "Time". The efficiency of our algorithm can be estimated by comparing this time with the enumeration time in column T_G of the same families of designs. These times are those taken by Gibbons's algorithm [7] on an IBM 370/165, executing approximately 3,000,000 operations per second. In many cases we have a great saving in time despite the fact that Gibbons was using a computer that was approximately ten times faster, and also restricted himself to the enumeration of designs without repeated blocks. Information about the families of the constructed BIBDs can be found in the papers listed in the "References" column. We assume that BIBDs for which this column is empty are first enumerated in our work.

Table 1

(v,b,r,k,λ)	$\|S\|$	Time(sec.)	T_G(sec.)	References
(6,10,5,3,2)	1	0.22	0.19	[4],[7]
(6,20,10,3,4)	4(1)	0.40		
(6,30,15,3,6)	6(0)	1.3		
(6,40,20,3,8)	13(0)	4.2		
(6,50,25,3,10)	34(0)	6.0		
(7,7,3,3,1)	1	0.23		[4]
(7,14,6,3,2)	4(1)	0.93		
(7,21,9,3,3)	10(1)	4.27		
(7,28,12,3,4)	35(1)	15		
(7,35,15,3,5)	108(1)	45		
(7,42,18,3,6)	417(0)	130		
(8,14,7,4,3)	4	1.67	3.38	[7],[9],[10]
(8,28,14,4,6)	≥578(?)	30 min.		
(9,12,4,3,1)	1	0.39		[4]
(9,24,8,3,2)	36(13)	62.0		
(9,36,12,3,3)	?(330)	4.5 hr.		
(9,18,8,4,3)	11	17.19	52.31	[7]
(10,30,9,3,2)	960(394)	5.2 hr.		
(10,15,6,4,2)	3	3.34	4.0	[7],[11]
(10,18,9,5,4)	21	71.1	280	[7],[12],[13]
(11,11,5,5,2)	1	0.63	1.61	[4],[7],[14],[15]
(11,22,10,5,4)	3337(?)	9 hr.		
(12,22,11,6,5)	601	4 hr.		
(13,26,6,3,1)	2	42.2		[4]
(13,13,4,4,1)	1	0.52	2.50	[4],[7]
(15,35,7,3,1)	80	3.4 hr.		[7],[16],[17]
(15,15,7,7,3)	5	34.8	127	[7],[10]
(16,20,5,4,1)	1	1.33	7.20	[4],[7]
(16,16,6,6,2)	3	48.15	45.85	[7],[15]
(19,19,9,9,4)	6	290	2160	[7],[13]
(21,21,5,5,1)	1	15.3		[4]
(23,23,11,11,5)	≥766	12 hr.		
(25,30,6,5,1)	1	123		[4]
(31,31,6,6,1)	1	9.5 min.		[4]

9. A generalization of the algorithm

Successively introducing some restrictions (§2), we chose BIBDs from the set of all incidence systems an we constructed an algorithm for the constructive enumeration of BIBDs. Here, by removing the introduced restrictions, we shall generalize it for arbitrary incidence systems.

First of all we note that the parameters r_i, k_j, and λ_{ps} were not used in the determination of canonicity of the incidence matrix (§2). Hence, for any introduced generalization $f \to F$, proposition 4.2 can be used as well as all results from §7 (by $f \to F$ we denote some generalization meaning that the previously used value f is replaced by a set of some values F).

a) $k \to \bar{K}_v$. It is easy to remove the restriction for the number of ones in the columns. Thus we shall pass to the consideration of the regular pairwise balanced designs with $k_i \in \bar{K}_v$ $(1 \leq i \leq b)$, where $\bar{K}_v$ is a nonempty set of integers. If $\bar{K}_v = \{0,1,2,...,v\}$, it is sufficient to search as before, but without using w-inadmissibility of some solutions of the system (3.2). The w-inadmissibility of these solutions was implied by the nonexistence of a completion such that each column of matrix A contained exactly k ones. By the same reason the "pumping" and the method of selected descendants cannot be used.

b) $\lambda \to \Lambda$. The enumeration of regular block designs which are not pairwise balanced is a more complicated problem. In this case at each search stage the system (3.2) is to be solved several times with every fixed right hand side, by substituting in the first equation the next number λ_i from Λ instead of λ. Thus we get all possible projections of the last rows on the partition N^w. The search on χ^w is performed as before.

c) $r \to \bar{R}$. We can enumerate the nonregular incidence systems by replacing $\chi^0 = r$ by $\chi^0 = R$.

d) $I \to S$. In §2, the special predicate S was introduced for the enumeration of incidence systems whose properties cannot be formulated in terms of dimensions v, b and sets $\bar{R}$, $\bar{K}_v$ and Λ. For BIBDs, $S=I$, where I is the identically true predicate. If $S \neq I$, then for solving some concrete problems its extension S_w is to be constructed and verified on the nodes of the tree U. Let $A^w(X_w^{w+1})$ be the matrix in which row w is constructed in accordance with the vector X_w^{w+1}. Then if $S(A^w(X_w^{w+1}))$ is not true, the vector X_w^{w+1} must be excluded from χ^{w+1} as it is inadmissible.

Table 2 presents the results of constructive enumeration of 3-designs.

Frequently the extension S_w cannot be effectively constructed. Therefore the truth of S must be verified on all fully completed matrices. This happens mainly for "global" properties of the incidence systems. Examples of this are constructive enumeration problems for group divisible designs [4], connected graphs, and Hamiltonian graphs. It is also difficult to construct effective

Table 2				
Parameters	$\|S\|$	Time(sec.)	T_G(sec.)	Reference
S(1;3,4,8)	1	0.25	0.91	[7]
S(2;3,4,8)	4(1)	3.05		
S(3;3,4,8)	10(1)	15.69		
S(4;3,4,8)	31(1)	40 min.		
S(1;3,4,10)	1	3.69		
S(3;3,5,10)	7	69.41	86.41	[7]
S(2;3,6,12)	1	6.18		
S(3;3,8,16)	5	212		

extensions if the predicate S describes some properties of the automorphism group (for example, transitivity) of enumerated objects.

In [19], using the described constructive enumeration technique, we constructed all regular edge- but not vertex-transitive graphs with at most 28 vertices, and we proved the nonexistence of such graphs with 30 vertices. The latter answers Folkman's question (4.2) from [20].

References

[1] A.V. Ivanov, "The construction and analysis of combinatorial block designs on the computer" (Russian), Autoreferat of thesis, MFTI, Moscow, 1980

[2] A.V. Ivanov and I.A. Faradžev, "Constructive enumeration of combinatorial block designs" (Russian), in *Algorithmic Researches in Combinatorial Theory*, "Nauka", Moscow, 1978, pp. 118-126.

[3] H. Hanani, "Balanced incomplete block designs and related designs", *Discrete Mathematics* 11 (1975) 255-369.

[4] M. Hall Jr., *Combinatorial Theory*, Blaisdell, Waltham MA, 1967.

[5] I.A. Faradžev, "Constructive enumeration of combinatorial objects", in *Problèmes Combinatoires et Théorie des Graphes*, CNRS, Paris, 1977, pp. 131-135.

[6] V.A. Zaitchenko and I.A. Faradžev, "The algorithm for testing canonicity of incidence systems" (Russian), in *Algorithmic Researches in Combinatorial Design Theory*, "Nauka", Moscow, 1978, pp. 126-129.

[7] P.B. Gibbons, "Computing techniques for the construction and analysis of block designs", Technical Report 92, Department of Computer Science, University of Toronto, 1976.

[8] P.B. Gibbons, R.A. Mathon, and D.G. Corneil, "Computing techniques for the construction and analysis of block designs", *Utilitas Mathematica* 11 (1977) 161-192.

[9] H.D. Gronau, "Über nichtisomorphe elementare blockwiederholungsfreie 2-(8,4,λ) Blockpläne I", *Rostocker Mathematisches Kolloquium (1979) 59-66.*

[10] H.K. Nandi, "A further note on nonisomorphic solutions of incomplete designs", *Sankhya* 7 (1946) 313-316.

[11] H.K. Nandi, "Enumeration of nonisomorphic solutions of BIBDs", *Sankhya* 7 (1946) 305-312.

[12] D.A. Preece, "Incomplete block designs with $v=2k$", *Sankhya* A29 (1967) 305-316.

[13] V.N. Bhat, "Nonisomorphic solutions of some BIBDs", *Journal of Combinatorial Theory* 12 (1972) 217-224.

[14] P.J. Cameron, "Biplanes", *Mathematische Zeitschrift* 131 (1975) 85-101.

[15] Q.M. Hussain, "On the totality of the solutions for the symmetrical incomplete block designs with $\lambda = 2$, $k = 5$ or 6", *Sankhya* 6 (1945) 204-208.

[16] F.N. Cole, A.S. White and L.D. Cummings, "Complete classification of triad systems on fifteen elements", *Memoirs of the National Academy of Science (U.S.A.)* 14 (1925) 1-89.

[17] M. Hall Jr. and J.D. Swift, "Determination of Steiner triple systems of order 15", *Mathematical Tables and Other Aids to Computation* 9 (1955) 146-156.

[18] N.L. Biggs, *Finite Groups of Automorphisms,* Lecture Notes in Mathematics 6, London Mathematical Society, Cambridge University Press, Cambridge UK, 1971.

[19] A.V. Ivanov, "On the regular line-symmetric, but not vertex-symmetric, graphs" (Russian), in *Permutation Groups and Combinatorial Objects,* Preprint 82.14, Institute of Mathematics AN USSR, 1982, pp. 38-51.

[20] J. Folkman, "Regular line-symmetric graphs", *Journal of Combinatorial Theory* 3 (1967) 215-232.

Annals of Discrete Mathematics 26 (1985) 247–274

Construction Procedures for t-designs and the Existence of New Simple 6-designs

Earl S. Kramer, David W. Leavitt and Spyros S. Magliveras

Department of Mathematics and Statistics, and
Department of Computer Science
University of Nebraska
Lincoln, Nebraska 68588
U.S.A.

Abstract

We describe procedures for finding t-designs with prescribed automorphism groups and apply these methods to finding t-designs on 20 points with either $PGL_2(19)$ or $PSL_2(19)$ as an automorphism group. We produce two non-isomorphic simple 6-designs with parameters 6-(20,9,112) and automorphism group $PSL_2(19)$. It has been previously shown that if $q < 19$, simple 6-designs on $v = q + 1$ points do not exist with automorphism group $PSL_2(q)$. Hence $v = 20$ is the smallest $v = q + 1$ where simple 6-designs occur with automorphism group $PSL_2(q)$.

1. Introduction

A t-design, or $t-(v,k,\lambda)$ design, is a pair (X,B) where B is a system of k-sets (called blocks) from a v-set X such that each t-set from X is in exactly λ blocks of B. A t-design is called *simple* if no block of B is repeated and *trivial* if each k-subset of X occurs precisely m times in B. In this paper we are interested primarily in nontrivial, simple t-designs. A necessary condition for the existence of a $t-(v,k,\lambda)$ design is that $\lambda\binom{v-i}{t-i} \equiv 0 \pmod{\binom{k-i}{t-i}}$ for $i = 0,1,2,\ldots,t$. In fact, Wilson (1973) showed that given v, k, t with $0 < t < k < v$, there is a constant $N(t,k,v)$ such that $t-(v,k,\lambda)$ designs exist for all $\lambda > N(t,k,v)$, where λ satisfies the above necessary conditions. A major problem is to find the minimum value for $N(t,k,v)$ and also to determine when simple, nontrivial t-designs occur. Success in finding simple t-designs for $t \geq 4$ has been limited. A good survey of results on t-designs is provided by A. Hedayat and S. Kageyama (1980). Briefly, there are a small number of infinite families of simple 4- and 5-designs, and only finitely many Steiner systems (t-designs with $\lambda=1$) for $t = 4$, or 5. Only recently have simple 6-designs been shown to exist. In 1982, S. Magliveras and D. Leavitt found the first simple 6-designs with parameters 6-(33,8,36). Magliveras focused his efforts on the unique 4-homogeneous, non set-transitive group $P\Gamma L_2(32)$ and Leavitt developed a new, powerful search procedure that uses the available information much more efficiently than before. In 1983, E. Kramer, D. Leavitt and S. Magliveras found the second set of 6-designs with parameters 6-(20,9,112) and automorphism group $PSL_2(19)$. Kramer (1975) had previously ruled out simple 6-designs on $v = 17$ points using $PSL_2(16)$, and on $v = 18$ points using $PSL_2(17)$. Hence, $PSL_2(19)$ on $v = 20$ points was an obvious situation to explore. In the following sections we describe our procedures for finding t-designs and apply them to the situation on $v = 20$ points with either $PGL_2(19)$ or $PSL_2(19)$ as an automorphism group.

2. Preliminaries.

A group action $G|X$ induces an action of G on the collection X_k of k-subsets of X for each $k \leq v = |X|$. Let $\rho = (\rho(0),\rho(1),...,\rho(v))$ be the vector whose k^{th} entry is the number of G-orbits on X_k. The entries $\rho(k)$ are easily given by the Frobenius-Cauchy-Burnside theorem; that is,

$$\rho(k) = [\textit{number of } G-\textit{orbits on } X_k] = |G|.^{-1} \sum_{g \in G} \theta_k(g)$$

where $\theta_k(g)$ is the number of k-subsets of X fixed by $g \in G$. A k-subset K of X is fixed by an element g of cycle type $1^{m_1}2^{m_2} \cdots n^{m_n}$ if and only if K is the union of cycles of g , Hence,

$$\theta_k(1^{m_1}2^{m_2}...n^{m_n}) = \sum_{(a_1,...,a_n)} \prod_{i=1}^{n} \binom{m_i}{a_i}$$

where the sum is taken over all non-negative integer vectors $(a_1,...,a_n)$ such that $\sum_{i=1}^{n} i.a_i = k$.

If A and B are k-subsets of X, in general it is a non-trivial task to decide whether A and B are in the same G-orbit of X_k. As we shall soon see, it is necessary to make many such decisions in the process of investigating the existence of t-designs with a prescribed group of automorphisms. We accomplish this by relying on 'invariant' functions.

Let $G|X$ be a group action and let R^{X_k} be the collection of all functions from X_k into a set R. The induced action $G|X_k$ is extended to R^{X_k} by $f^g(A) = f(A^g)$ for $A \in X_k$, $g \in G$. A function $f \in R^{X_k}$ fixed by all elements of G is called G-invariant, or simply *invariant*. Suppressing G and X, we denote the collection of all invariant functions in R^{X_k} by $\Omega_k(R)$. Note that when R is a ring, $\Omega_k(R)$ is a free R-module of rank $\rho(k)$.

By the *rank* of $f \in \Omega_k(R)$, we mean the number $r(f)$ of distinct values taken by f in R, thus, $r(f) = |f(X_k)|$. A function $f \in \Omega_k(R)$ is called a ***discriminator*** if $r(f) = \rho(k)$. We observe that a function $f : X_k \to R$ is G-invariant if and only if f is constant on the G-orbits in X_k. Thus, if f is invariant, $A, B \in X_k$, and $f(A) \neq f(B)$ then A and B are not in the same orbit. Moreover, for f a discriminator, $A, B \in X_k$ are in the same orbit if and only if $f(A) = f(B)$. If $f : X_k \to R_1$, $g : X_k \to R_2$ are functions then the cartesian product $f \times g$ is defined by $(f \times g)(A) = (f(A), g(A))$. The following statement is easy to see:

Lemma 1 If $f \in \Omega_k(R_1)$ and $g \in \Omega_k(R_2)$, then $f \times g \in \Omega_k(R_1 \times R_2)$, and $r(f \times g) \geq \max\{r(f), r(g)\}$.

If f and g are invariant functions we say that f *dominates* g, denoted by $f > g$, if $r(f \times g) = r(f)$. We say that f is *equivalent* to g, $f \sim g$, if $f > g$ and $g > f$. Frequently the result of taking the product of two invariant functions f and g results in a function strictly dominating both f and g. This allows us to construct discriminators by iteratively taking cartesian products of invariant functions of small rank. The *efficiency* of an invariant function $f \in \Omega_k(R)$ is defined to be the ratio $\eta(f) = r(f)/\rho(k)$, thus, an invariant function is a discriminator if and only if it has efficiency 1.

3. G-fused Incidence Matrices

In 1976, Kramer and Mesner elucidated the role of certain matrix invariants associated with a group action $G|X$. Roughly speaking such a matrix is the result of fusing under G the incidence matrix between X_t and X_k where incidence is set inclusion. These matrices contain, in a concise way, all the relevant information for investigating the existence of t-designs with automorphism group G. We proceed to introduce these matrices.

For $1 \leq t < k < v = |X|$, let $\{\Delta_i^{(t)}: i=1,\ldots,\rho(t)\}$, $\{\Delta_j^{(k)}: j=1,\ldots,\rho(k)\}$, be the collections of orbits of G on X_t and X_k respectively. For a fixed member T of $\Delta_i^{(t)}$ the number $a_{ij}(T)$ of members $K \in \Delta_j^{(k)}$ such that $T \subset K$ is independent of the choice of $T \in \Delta_i^{(t)}$, hence we may write $a_{ij} = a_{ij}(T)$. We define the $\rho(t)$ by $\rho(k)$ matrix $A_{t,k} = A_{t,k}(G)$ by: $A_{t,k} = (a_{ij})$.

Dually, for a fixed member K of $\Delta_j^{(k)}$, the number $b_{ij}(K)$ of members T of $\Delta_i^{(t)}$ such that $T \subset K$ is independent of the choice of K in $\Delta_j^{(k)}$, and we define the matrix $B_{t,k} = B_{t,k}(G)$ by $B_{t,k} = (b_{ij})$. For $k = 1,\ldots,v$, let $L_k = (L_k(1),\ldots,L_k(\rho(v)))$ be the vector of orbit lengths of G on X_k, that is $L_k(i) = |\Delta_i^{(k)}|$. For the pair of orbits $\Delta_i^{(t)}$ and $\Delta_j^{(k)}$ the entries a_{ij} and b_{ij} can be thought of as the degrees of a regular bipartite graph with vertex set $\Delta_i^{(t)} \cup \Delta_j^{(k)}$ where $T \in \Delta_i^{(t)}$ is joined to $K \in \Delta_j^{(k)}$ if and only if $T \subset K$.

Finally, we introduce a third family of matrices related to $G \mid X$. Let K be a fixed member of $\Delta_i^{(k)}$ and let c_{ij} be the number of elements of $\Delta_j^{(k)}$ that intersect K in exactly t points. We define the $\rho(k)$ by $\rho(k)$ matrix $C_{t,k} = C_{t,k}(G)$ by $C_{t,k} = (c_{ij})$.

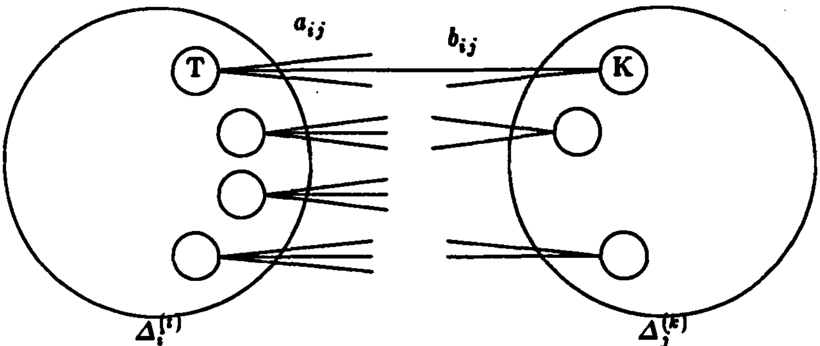


Figure 1

We mention here, some useful properties of the matrices $A_{t,k}$, $B_{t,k}$, $C_{t,k}$ and the orbit length vectors L_t. Statement (iv) which is an easy consequence of (iii) was first observed by Leo G. Chouinard. Statement (v) was discovered by D. Kreher and independently by D. Leavitt.

Lemma 2 Let $A_{t,k}$, $B_{t,k}$, $C_{t,k}$, L_t be as defined above.

(i) If $t \le s \le k$ then $A_{t,k} = \binom{k-t}{k-s}^{-1} A_{t,s} A_{s,k}$

(ii) $A_{t,k}$ has constant row sums $\binom{v-t}{k-t}$

(iii) $L_t(i) A_{t,k}(i,j) = L_k(j) B_{t,k}(i,j)$

(iv) $\binom{k}{t} L_k = L_t A_{t,k}$

(v) $$C_{t,k} = B_{t,k}^T A_{t,k} - \sum_{i=t+1}^{k} \binom{i}{t} C_{i,k}$$

Proof: Properties (i) - (iv) are immediate. We sketch a proof of (v). Note that the $(i,j)^{th}$ entry of $B_{t,k}^T A_{t,k}$ is the number of triples (K,T,J), K a fixed member of $\Delta_i^{(k)}$, $J \in \Delta_j^{(k)}$ and $T \subset K \cap J$, with $|T| = t$. On the other hand for $r \ge t+1$ $\binom{r}{t} C_{r,k}(i,j)$ is the number of triples (K,T,J), K and J as above, where $T \subset K \cap J$, $|K \cap J| = r$. Hence the formula.

Note that the above lemma allows one to compute $\{A_{t,k} : t < k\}$ and $\{B_{t,k} : t < k\}$ from L_1 and $\{A_{i,i+1} : i = 1,\dots,[(v+1)/2]\}$.

Let $A_{t,k}$ be defined above for some pair t, k, $1 \le t < k < v$. Suppose furthermore that there exists a collection of columns $j_1,...,j_q$ of $A_{t,k}$, corresponding to the G-orbits of k-sets $\Delta_{j_1}^{(k)}, \ldots, \Delta_{j_q}^{(k)}$, whose sum is the vector $(\lambda,\lambda, \ldots, \lambda)^T$. This simply means that the union B of orbits $\Delta_{j_1}^{(k)}, \ldots, \Delta_{j_q}^{(k)}$ is a collection of k-subsets of X with the property that any t-subset of X occurs in exactly λ members of B. Hence, B is a G-invariant $t-(v,k,\lambda)$ design. Moreover, if the columns $j_1,...,j_q$ are distinct, no k-sets repeat, so B is a simple design. The converse is easily seen to hold, hence we have the following result.

Theorem 3 [Kramer and Mesner, (1976)]. There exists a $t-(v,k,\lambda)$ design with the underlying point set X, $|X| = v$, and with G a group of automorphisms if and only if there exists a solution U to the matrix equation $AU = \lambda J$, where $A = A_{t,k}$, U is a $\rho(k)$-dimensional vector of non-negative integral entries, J is the $\rho(t)$-dimensional vector of all 1's, and λ a positive integer. The t-design is simple if and only if U is a 0-1 vector.

We now proceed to investigate the relationship between $A_{t,k}(H)$ and $A_{t,k}(K)$ when H and K are subgroups of G with $H \le K \le G$. Let $A = (a_{ij})$ be an $m \times n$ matrix with non-negative integral entries and constant row sums. A pair $P = (\pi_1,\pi_2)$ where $\pi_1 = \{D_i\}_{i=1}^{u}$ is a partition of $\{1,...,m\}$ and $\pi_2 = \{F_j\}_{j=1}^{w}$ is a partition of $\{1,...,n\}$ is called a *tactical fusion* of $(\{1,...,m\},\{1,...,n\})$ for A if $1 \le i \le u$, $1 \le j \le w$, $x, y \in D_i$ implies that

$$\sum_{q \in F_j} a_{x,q} = \sum_{q \in F_j} a_{y,q} = \bar{a}_{i,j}$$

We set $A[P] = (\bar{a}_{ij})$. The *tactical domain* of A denoted by $D(A)$ is the set of all tactical fusions for A. If $B = A[P]$ for some $P \in D(A)$ we say that B *covers* A and write $A \le B$.

Suppose that $G|X$ is a group action and that $1 \le t < k \le v = |X|$. Let A be the incidence matrix between X_t and X_k where incidence is set inclusion. The following proposition is easy to show:

Proposition 4 Let H and K be the subgroups of G so that $H \le K$. Let π_1, π_2 be the systems of H-orbits on X_t, X_k and σ_1, σ_2 the systems of K-orbits on X_t, X_k respectively, then

(i) $P = (\pi_1,\pi_2)$ is a tactical fusion for A

(ii) $A[P] = A_{t,k}(H)$

(iii) $S = (\sigma_1,\sigma_2)$ induces a tactical fusion of P for $A[P]$

(iv) $\{A[P]\}[S] = A_{t,k}(K)$

(v) $A_{t,k}(H) \leq A_{t,k}(K)$

4. An Algorithm for Computing Incidence Matrices

Direct computation of the matrices $A_{k-1,k}$ by means of actually computing and storing orbits $\{\Delta_i^{(k-1)}\}$ and $\{\Delta_j^{(k)}\}$ is very inefficient in terms of both machine time and space. It suffices to note that the number of orbits of G on X_r is bounded below by $\binom{|X|}{r}/|G|$ and that most orbits are regular, that is of length $|G|$. We proceed to describe a much better algorithm for computing $A_{k-1,k}$. Here, we assume that we have representatives of each of the $\rho(k-1)$ G-orbits on $(k-1)$-sets, say $T(k-1,1),\ldots,T(k-1,\rho(k-1))$, and the corresponding vector of orbit lengths L_{k-1}. We also assume that a sequence $(f_1,f_2,\ldots,f_n)$ of functions in $\Omega_k(Z)$ is made accessible to the algorithm so that for some $m \leq n$, $f_1 \times f_2 \times \cdots \times f_m$ is a discriminator. The algorithm proceeds to compute $A_{k-1,k}$, representatives of each of the $\rho(k)$ orbits of G on X_k, and the vector of orbit lengths L_k. The algorithm makes r passes to complete the process, where r is the least integer such that $f_1 \times \cdots \times f_r$ is a discriminator.

Algorithm 5

1. Initialize F to an $n \times (\rho(k-1)(v-k+1))$ zero matrix
2. For $m = 1$ to n, step $= 1$
3. Set function f equivalent to function f_m
4. Set $indx = 0$
5. For $i = 1$ to $\rho(k-1)$, step $= 1$
6. Compute the complement $Y_i = X \setminus T(k-1,i)$
7. For $j = 1$ to $(v-k+1)$, step $= 1$
8. Set $indx = indx + 1$
9. Set $q_j = j^{th}$ element of Y_i
10. Compute $T^+ = T(k-1,i) \cup \{q_j\}$
11. Set $F(m,indx) = f(T^+)$
12. Next j ; Next i
13. Compute $R =$ [the number of distinct columns of F]
14. If $R = \rho(k)$ then go to **Step 18**
15. Next m

16. Store F on mass-storage device for later use
    Print: 'Discrimination was not achieved. Increase Pool of invariant functions ...'
17. Stop
18. Convert information in F to $A_{k-1,k}$, and print $A_{k-1,k}$.
19. Stop

We proceed to describe some easily computable invariant functions.

4.1. Anchor Sets

Let A be a fixed subset of X which we shall call an *anchor set*. We describe an invariant function $f_A \in \Omega_k(Z^{k+1})$ as follows: Begin by calculating the orbit $\Delta = A^G = \{A_1,...,A_s\}$. Now, for any $B \in X_k$, we define the *frequency vector* of B relative to the anchor set A to be $f_A(B) = (f_0,f_1,...,f_k)$ where f_i is the number of members A_j of Δ intersecting B in exactly i points. These invariant functions f_A appear to be of low efficiency, when $|A|$ is small and k is of size close to $[v/2]$. The efficiency of f_A improves as the size of A increases to $[v/2]$. Typically the cartesian product of a few judiciously chosen f_{A_i} has produced a discriminator.

4.2. Taxonomy 1

Suppose that G contains an element π which is represented on X as a regular permutation of type s^m. Then the cyclic group $<\pi>$ has a system $\gamma = \{C_1,...,C_m\}$ of orbits on X, each of size s; that is, γ is a regular partition of X. Let $F = \gamma^G$ be the orbit under G of the partition γ; Thus, F contains all partitions of type $\gamma^g = \{C_1^g,...,C_m^g\}$, $g \in G$. Now let B be any member of X_k. If $\delta = \{D_1,...,D_m\} \in F$, we compute the frequency vector of B relative to the partition δ by: $f(\delta,B) = (f_0,f_1,...,f_q)$ where f_i is the number of blocks of δ intersecting B in exactly i points, $q = \min\{s,k\}$. As δ runs through the orbit of partitions in F, $f(\delta,B)$ runs through a specific set of distinct frequency vectors. We tabulate the frequencies with which the distinct frequency vectors appear, and obtain a frequency vector of frequency vectors $\mu_\pi(B)$. The function μ_π is clearly invariant, apparently of high efficiency, and it appears that the efficiency in discriminating G-orbits on X_k increases with k in $[1,v/2]$. In several instances, η turns out to be $1-(1/\rho(k))$.

4.3. Taxonomy 2

The next procedure in computing invariant functions is motivated by the matrices $B_{t,k}$. Suppose that for some $t < k$ we have been successful in obtaining a discriminator function ϕ_t. Let $\{\Delta_i^{(t)} : i=1,...,\rho(t)\}$ be the orbits of G on X_t, and let B be an arbitrary k-subset of X. Now, consider the vector $\nu_t(B) = (f_1, f_2, ..., f_{\rho(t)})$ where f_i is the number of t-subsets of B which belong to $\Delta_i^{(t)}$. To compute $\nu_t(B)$, we run through the $\binom{k}{t}$ t-subsets T of B, each time determining the orbit $\Delta_i^{(t)}$ in which T falls by computing $\phi_t(T)$.

5. Leavitt's Algorithm

A crucial step in deciding whether a given group action $G|X$ supports simple $t-(v,k,\lambda)$ designs is the investigation of existence of 0-1 solutions to the matrix equation $AU = \lambda J$. An upper bound to the time complexity for the problem is $2^{\rho(k)}$. Since in the $PSL_2(q)$ case $\rho(k)$ is asymptotically $\binom{v}{k} / |G| \sim cq^{k-3}$, c a constant, we see that backtrack is hopeless with complexity $2^{(q^{k-3})}$. Efforts were made to adopt an optimization algorithm for integral bivalent problems by Egon Balas (1975) but we were unsuccessful in obtaining results with it. Balas' algorithm on the other hand yields one, optimal solution, if any, with respect to a predefined objective function. We were interested in all solutions with the null objective function. We still intend to study the feasibility of Balas' algorithm for our design searches.

In what follows we discuss a procedure for obtaining all 0-1 solutions to the integral matrix equation $AU = B$. The procedure can be viewed as solving by subspaces and involving space-time tradeoffs. To make the presentation of the method easier we assume that the machine used has unlimited storage. In actual practice a user will make modifications to adopt the process to his own machine constraints.

We assume that A and B are set up as $m \times n$ and $m \times r$ integer matrices respectively. In addition we set up an $n \times r$ zero matrix for U, where solutions are to be accumulated, and introduce a $1 \times n$ vector F which is used to flag columns of A and rows of U. None of the four matrices A, B, U, F are static in the sense that their dimensions will change during the procedure. In particular, r will fluctuate considerably during execution as it corresponds to the number of accumulated potential solutions in the search. We proceed to discuss elementary operations for the procedure.

Gauss Operations

$G[i]$ Divide the i^{th} row of $[A,B]$ by the greatest common divisor of the elements of this row.

$G[i,j]$ Interchange rows i and j of $[A,B]$.

$G[\alpha;i,j]$
Add an integer multiple of row i to row j of $[A,B]$. The multiplier is α.

Expansion Operations

$E[p;i]$ Let p be a positive integer and $P = [A,B] =$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & ; & b_{1,1} & b_{1,2} & \cdots & b_{1,r} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & ; & b_{2,1} & b_{2,2} & \cdots & b_{2,r} \\ \cdot & \cdot & \cdots & \cdot & & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & & \cdot & \cdot & \cdots & \cdot \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & ; & b_{m,1} & b_{m,2} & \cdots & b_{m,r} \end{bmatrix}$$

Catenate an $(m+1)^{st}$ row to P, $(a_{m+1,1}, a_{m+1,2}, \ldots, a_{m+1,n}; \quad b_{m+1,1}, b_{m+1,2}, \ldots, b_{m+1,r})$ where $a_{m+1,j}$ is a reduced residue class representative of $a_{i,j}$ modulo p, and $b_{m+1,j} \equiv b_{i,j} \pmod p$. If S is the 0-1 span of $\{a_{m+1,j}\}_{j=1}^{n}$, then for each $s \in S$, $s \equiv b_{i,j} \pmod p$ catenate a column $(b_{1,j}, \ldots, b_{m,j}, s)^T$ to $[A,B]$ and a column $(u_{1,j}, \ldots, u_{n,j})^T$ to U.

$E[i]$ If the elements of the i^{th} row of A are contained in {-1,0,1}, then catenate an $(m+1)^{st}$ row to P with

$$a_{m+1,j} = \begin{cases} 1 & \text{if } a_{i,j} = -1 \\ 0 & \textit{otherwise} \end{cases}$$

and $b_{m+1,j} = b_{i,j}$. If S is the 0-1 span of $\{a_{m+1,j}\}_{j=1}^{n}$ then for each $s \in S$, $s > -b_{i,j}$, catenate a column $(b_{1,j}, \ldots, b_{m,j}, s)^T$ to B and catenate a column $(u_{1,j}, \ldots, u_{n,j})^T$ to U.

Contraction Operations

$C1[i]$ If $b_{i,j}$ is not in the 0-1 span of $\{a_{i,k}\}_{k=1}^{n}$ then remove the j^{th} columns of B and U.

$C2[i]$ If the greatest common divisor of $\{a_{i,k}\}_{k=1}^{n}$ does not divide $b_{i,j}$ then delete the j^{th} columns of B and U.

Resolution Operation

$R[i]$ If the i^{th} row of A has all zeroes except that $a_{i,j} = 1$ then we can substitute row i of B into row $F(j)$ of U, delete the i^{th} row of $[A,B]$, delete the j^{th} column of A, and delete the j^{th} entry of F.

Many of the procedures can be combined and follow each other naturally, such as $C2[i]$ and $G[i]$. $E[i]$ was originally a combination of $E[2;i]$, $G[1;m+1,i]$, $C2[i]$, $G[-1;i,m+1]$, etc. Any procedure which changes the number of rows or columns of A or B must update the value of m, n, or r correspondingly.

Algorithm 6

1. Initialize m, n, and r as scalars
2. Enter A and B, an $m \times n$ and $m \times r$ matrix respectively
3. Initialize U to an $n \times r$ zero matrix
4. Initialize $F = (F(1),...,F(n))$ with $F(i) = i$
5. Set $i = 1$
6. While $i < n$ do
7. Set $h = i + 1$
8. While $h \leq n$ do
9. Set boole = true
10. For $j = 1$ to m, step = 1
11. Set boole = boole $\wedge$ $(a_{i,j} = a_{h,j})$
12. Next j
13. If not boole then go to **Step 22**
14. Set $U1 = U$, $B1 = B$
15. For $j = 1$ to r, step = 1
16. Set $u1_{h,j} = 1$

17. For $k = 1$ to m, step $= 1$
18. Set $b1_{k,j} = b1_{k,j} - a_{k,h}$
19. Next k ; next j
20. Set $B = [B;B1]$, $U = [U;U1]$, $n = n - 1$
21. Delete $F(h)$ and h^{th} column of A
22. Set $h = h + 1$
23. End while
24. Set $i = i + 1$
25. End while
26. Compute s = [the index of the first row of A which is not a 0-1 vector]
27. Set index = 0; numodd = n; $i = s$
28. While $i \leq m$ do
29. $C2[i]$; $G[i]$; $C1[i]$
30. Compute q = [number of odd entries in i^{th} row of A]
31. If $q \geq$ numodd then go to **Step 33**
32. Set index = i; numodd = q
33. Set $i = i + 1$
34. End while
35. E[2;index] ; E[-1;m,index] ; $C2$[index] ; G[index] ; $G[m,s]$
36. Set $i = 1$
37. While $i \leq m$ do
38. If $i = s$ then go to **Step 40**
39. $G[-a_{i,j};s,i]$
40. $C1[i]$
41. Set $i = i + 1$
42. End while
43. Set $t = m$
44. While $i < s$ do
45. Set count = 0
46. For $j = 1$ to n, step $= 1$

47. If $a_{i,j} \neq 0$ then set count = count + 1
48. If $a_{i,j} \neq -1$ then next j
49. $E[i]$; $G[1;m,i]$
50. Next j
51. If count $\neq 1$ then set $i = i + 1$ else set $s = s - 1$
52. $R[i]$
53. End while
54. If $n > 0$ then go to **Step 26**
55. if $r = 0$ then print 'no solutions' else print U
56. Stop

6. The simple t-designs from $PGL_2(19)$ and $PSL_2(19)$ with $3 \leq t \leq 5$

In what follows we let $X = GF(19) \cup \{\infty\} = \{1,2,...,19,20\}$ where we identify ∞ with 20 and 19 with the zero in $GF(19)$. The group $PSL_2(19)$, of order 10.19.18 = 3420, is generated by the two elements $\alpha : x \rightarrow x + 1$ and $\beta : x \rightarrow -1/x$. Then $PGL_2(19)$, of order 20.19.18 = 6840, is generated by α, β, and $\gamma : x \rightarrow -x$. In permutation form, $\alpha =$ (1 2 3 ... 18 19)(20), $\beta =$ (1 18)(2 9)(3 6)(4 14)(5 15)(7 8)(10 17)(11 12)(13 16)(19 20), and $\gamma =$ (1 18)(2 17)...(9 10)(19)(20). The group $PGL_2(19)$ is sharply 3-transitive on X, and $PSL_2(19)$ is 3-homogeneous on X. In Table 1 we list orbit representatives for each of the $PSL_2(19)$ orbits on X_k for $3 \leq k \leq 10$.

If a PSL orbit Δ is fixed by the outer automorphism γ then Δ is also a PGL orbit and we label it by using the same unsigned integer index for both groups. A PSL orbit Δ which is not fixed by γ is carried into another PSL orbit Δ^γ. The two PSL orbits Δ and Δ^γ fuse to produce the PGL orbit $\Delta \cup \Delta^\gamma$. We denote a pair of PSL orbits interchanged by γ by a pair of signed integer indices j^- and j^+. These fuse to produce orbit j of PGL. For example, for $k = 4$, PSL orbits 1^- and 1^+ are interchanged by γ and fuse into PGL orbit 1, while PSL orbit 2 is fixed by γ. Our notation depicts which PSL orbits are PGL orbits and which PSL orbits fuse in pairs to create PGL orbits.

From the matrix $A = A_{4,5}(PSL_2(19))$

	1	2	3	4	5^-	5^+
1^-	0	4	0	0	12	0
1^+	0	4	0	0	0	12
2	0	0	0	8	4	4
3	2	0	4	2	4	4
4	0	2	6	4	2	2

we obtain the matrix $A_{4,5}(PGL_2(19))$ from $A_{4,5}(PGL_2(19)) = A[S]$ where $S = (\sigma_1,\sigma_2)$ is the tactical fusion, with $\sigma_1 = \{\{1^-,1^+\}, \{2\},\{3\},\{4\}\}$ and $\sigma_2 = \{\{1\},\{2\},\{3\},\{4\},\{5^-,5^+\}\}$.

Note for example that in $A_{4,5}(PSL_2(19))$ the 2 by 2 submatrix corresponding to PSL orbits 1^-, 1^+ cn 4-sets and 5^-, 5^+ on 5-sets is $\binom{12\ \ 0}{0\ \ 12}$. The corresponding entry in the $A_{4,5}(PGL_2(19))$ is then the common row sum 12 of the submatrix. We quickly obtain $A_{4,5}(PGL_2(19))$ as

	1	2	3	4	5
1	0	4	0	0	12
2	0	0	0	8	8
3	2	0	4	2	8
4	0	2	6	4	4

In Table 2 we display the transposed $A_{t,k}$ matrices arising from $PSL_2(19)$ for $3 \le t < k \le 10$ and $t \le 7$, but not for $t = 7$ and $k = 8$. Implicitly, the $A_{t,k}$ matrices for $PGL_2(19)$ are also given for the same values of t and k, since the tactical fusion induced by $PGL_2(19)$ is completely specified.

Note that we have drawn lines in the $PSL_2(19)$ matrices to delineate all of the 1 by 1, 1 by 2, 2 by 1, or 2 by 2 submatrices that collapse to produce the appropriate entry in the $PGL_2(19)$ matrix.

In Table 3 we have indicated the λ's for which there exists a simple nontrivial $t-(v,k,\lambda)$ design for $PGL_2(19)$ or $PSL_2(19)$. Since any design with automorphism group $PGL_2(19)$ will also have the subgroup $PSL_2(19)$ as an automorphism group, the right hand column lists only those values of λ for which there is a $t-(v,k,\lambda)$ design with automorphism group $PSL_2(19)$ but not $PGL_2(19)$. In our notation we let λ be the minimal admissible value of $\underline{\lambda}$ for the fixed t, k, and $v = 20$. Further $\bar{\lambda} = \binom{v-t}{k-t}$ is the value of λ if one were to use all the k-subsets of X to form the trivial t-design. If (X,B) is a simple, nontrivial t-design, then $(X,X_k \backslash B)$ is also a t-design, so we mention solutions only for $\lambda \leq \bar{\lambda}$ where $\bar{\lambda}$ is the largest admissible value of λ less than or equal to $\lambda/2$.

When $t = 3$ and for each value of $k = 4,5,\dots,10$ it is elementary to check that we get nontrivial, simple 3-designs for precisely those values of λ listed in Table 3.

When $k = 5$ and $t = 4$ it is easy to check that there are no t-designs for either of the two groups. For $k = 6$, $t = 4$ a short search rules out any designs for $PGL_2(19)$, but there are several solutions for $PSL_2(19)$, all with $\lambda = 60$. One such solution is, for example, $(2^-,4,6,8^+,11,12,13^-$. The 5-designs for either group with $k = 6$ must be trivial since $\underline{\lambda} = \bar{\lambda}$.

When $k = 7$, $t = 4$ the first row of the matrix for both groups implies that 4 must divide λ, so that only $\lambda = 140$ or 280 are possible. We get 18 solutions in all for $\lambda = 280$ from $PGL_2(19)$. One such solution is (2,3,4,5,9,11,13,16,17). The designs for $t = 5$ or 6 are trivial since $\underline{\lambda} = \bar{\lambda}$.

For $k = 8$ and $t = 4$ or 5 we get the solutions exhibited in Table 3 with an example for each λ provided in Table 4. If $t = 6$ the first row of the matrix for $PSL_2(19)$ forces $\lambda \equiv 0,1 \pmod 3$ leaving $\lambda = 7,21,28,42$ as the only possibilities. Relatively easy searches exclude each such value of λ, so 6- or 7-designs do not occur here.

For $k = 9$, when $t = 4$ an easy check of orbit lengths shows that if $\lambda = 168x$ then $x \not\equiv 1 \pmod 3$. If $t = 5$ and $\lambda = 105x$ then similar reasoning shows that $x \not\equiv 2 \pmod 3$. The 4- and 5-designs otherwise exist and an example of each possible value of λ appears in Table 4. A short search for $PGL_2(19)$ rules out any t-designs for $t \geq 6$. For $PSL_2(19)$ there exist exactly two nonisomorphic 6-(20,9,112) designs which are not 7-(20,9,24) designs, and which we discuss in the next section. Here no t-designs for $t \geq 7$ exist for $k = 9$ with automorphism group $PSL_2(19)$.

For $k = 10$ the λ for 3-designs satisfies $\lambda \not\equiv 4 \pmod 5$ and this forces the corresponding λ for $t = 4$ and $t = 5$ to satisfy $\lambda \not\equiv 4 \pmod 5$. If $t = 4$ then row 1 of the matrix for $PGL_2(19)$ forces $\lambda \not\equiv 2 \pmod 6$. If $t = 5$ then consideration of possible orbit lengths for $PGL_2(19)$ when $\lambda = 21x$ forces $x \not\equiv 1 \pmod 3$. Examples of designs, for a situation where a design exists for a particular value of λ, are obtained by using unions and complements of unions of disjoint designs listed in Table 4. For example, if $t = 5$ and $\lambda = 1113$ we get a design which is the union of disjoint designs with λ values of 315, 378, and 420 respectively. Or if $t = 5$ and $\lambda = 1302$ we can get such a design as the complement of the union of disjoint designs with λ values of 315, 630, and 756, respectively.

For $k = 10$ and $t \geq 6$ we have ruled out designs having $PGL_2(19)$ as an automorphism group. This effort was greatly expedited by using the search procedure developed by D. Leavitt. In particular, the first author had ruled out all but 5 values of λ for $6-(20,10,\lambda)$ fixed by $PGL_2(19)$ and had estimated it would take several years of CPU time to eliminate just one of these remaining λ values by the backtrack procedure he was using.

For $k = 10$ and $t \geq 6$ when $PSL_2(19)$ is the automorphism group we have eliminated any designs for $t = 9$. We are still in the process of examining whether any 6-, 7-, or 8-designs can exist. For $t = 6$, by considering orbit lengths we must have $\lambda \not\equiv 3 \pmod 5$ and we have eliminated $6-(20,10,\lambda)$ designs for $\lambda < 140$ with $PSL_2(19)$ as automorphism group.

7. New simple 6-designs with automorphism group $PSL_2(19)$

The 6-(20,9,112) designs which we discovered are the smallest possible cases where simple 6-designs occur with $PSL_2(q)$ as an automorphism group on $v = q + 1$ points. The case $q = 13$ has been ruled out by several people including L. Chouinard and D. Kreher (private communication). E. Kramer (1975) established that there were no simple nontrivial 6-designs on $v = 17$ points using $PSL_2(16)$ and he also determined that there are no 6-designs on $v = 18$ points with automorphism group $PSL_2(17)$.

There are exactly four solutions to the matrix equation $A_{6,9}U = 112J$ and the vectors of orbit indices producing these solutions are S_1, S_1^*, S_2, S_2^* where

$S_1 =$ (1,2,3-,3+,5,16-,17-,18+,21-,23-,23+,24-,24+,26+,28+,30+,32-,32+)
$S_1^* =$ (1,2,3-,3+,5,16+,17+,18-,21+,23-,23+,24-,24+,26-,28-,30-,32-,32+)
$S_2 =$ (1,2,3-,3+,5,7,15,17-,17+,20+,21-,22-,25-,25+,28-,28+,30-,32+),
$S_2^* =$ (1,2,3-,3+,5,7,15,17-,17+,20-,21+,22+,25-,25+,28-,28+,30+,32-).

Theorem 7

There are exactly two nonisomorphic simple nontrivial 6-(20,9,112) designs with $PSL_2(19)$ as an automorphism group.

Proof: The outer-automorphism $\gamma \in PGL_2(19) \setminus PSL_2(19)$ interchanges S_1 with S_1^* and S_2 with S_2^* so that we have at most two non-isomorphic 6-(20,9,112) designs. Suppose that S_1 is isomorphic to S_2. Then there exists a permutation π in the symmetric group $\Sigma = \Sigma_{20}$ such that $S_1^\pi = S_2$. It is known that if $PSL_2(19) < H \leq \Sigma_{20}$ then, $H = \Sigma_{20}$, the alternating group A_{20} or $PGL_2(19)$. We also know that none of these overgroups of $PSL_2(19)$ preserve either S_1 or S_2. Then,

$$PSL = \Sigma_{(S_2)} = \Sigma_{(S_1^\pi)} = (\Sigma_{(S_1)})^\pi = PSL^\pi$$

where PSL is the particular $PSL_2(19)$ fixing S_1 and S_2. Thus, π normalizes PSL, and therefore, $\pi \in PGL_2(19)$ contrary to the fact that S_1 is not carried into S_2 under elements of $PGL_2(19)$.

In Table 5 we display the intersection numbers of the two 6-(20,9,112) designs S_1 and S_2. Here, if B is a block in an orbit constituent of design (X,B) we tabulate the number of blocks in B which intersect B in exactly j points. If U is a 0-1 solution to $AU = \lambda J$, corresponding to the simple design (X,B), the intersection numbers for (X,B) appear in the product $C_{t,k}U$.

The upper section of Table 5 gives this information for the design S_1 and the lower section for the design S_2.

Note that these intersection numbers provide an alternate proof that S_1 and S_2 are nonisomorphic. For example, there are blocks in S_1 that are disjoint from 20 other blocks of S_1, whereas no blocks in S_2 are disjoint from more than 19 other blocks of S_2.

8. Closing Remarks

The authors feel strongly that there exist simple non-trivial t-designs for arbitrarily large values of t. Further we conjecture that for any fixed value of t there is a q for which a simple, nontrivial t-design exists with $PSL_2(q)$ as its automorphism group.

One major difficulty in seeing what are the appropriate analogues of the t-designs for $t = 6$ that were found so far, is the very nontrivial problem of characterizing the orbits of $PSL_2(q)$ on X_k for general q and k. Note that solely group theoretic characterizations can not work since most orbits are regular. Clearly some invariants of a structural or geometric nature are needed.

Another major difficulty lies in finding nice examples for relatively small situations. Even in the case $v = 20$, with $PSL_2(19)$ as the automorphism group our search procedures have not completely finished all cases for the possible existence of t-designs for $6 \leq t \leq 8$ and $k = 10$. Hence improved algorithms are very much needed.

Acknowledgements

The authors wish to thank their colleagues Prof. D.M. Mesner, Prof. D.A. Klarner, Mr. D.L. Kreher, and especially Prof. L.G. Chouinard for useful and substantive discussions. The authors also wish to thank the University of Nebraska for support of this research.

References

[1] W.O. Alltop, "An infinite class of 4-designs", *Journal of Combinatorial Theory* 6 (1969) 320-322. MR 39 # 2656.

[2] W.O. Alltop, "An infinite class of 5-designs", *Journal of Combinatorial Theory* (A) 12 (1972) 390-395.

[3] E.F. Assmus Jr. and H.F. Mattson Jr., "New 5-designs", *Journal of Combinatorial Theory* 6 (1969) 122-151.

[4] E. Balas, *Bivalent Programming by Implicit Enumeration,* Encyclopedia of Computer Science and Technology, vol. 2, M. Dekker Inc., (1975) N.Y.

[5] R.D. Carmichael, *Introduction to the theory of groups of finite order,* Dover Publications, Inc., (1956) N.Y.

[6] P. Dembowski, *Finite Geometries,* Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 44, Springer Verlag, (1968) Berlin, N.Y. MR 38 # 1597.

[7] R.H.F. Denniston, "Some new 5-designs", *Bulletin of the London Mathematical Society* 8 (1976) 263-267.

[8] B.H. Gross, "Intersection triangles and block intersection numbers of Steiner systems", *Mathematische Zeitschrift* 139 (1974) 87-104. MR 50 # 9614.

[9] M. Hall Jr., *Combinatorial Theory* Blaisdell Publishing Company, Waltham, Mass., Toronto, Ont., London. MR 37 # 80.

[10] M. Hall Jr., *The Theory of Groups* Macmillan, New York.

[11] H. Hanani, "On quadruple systems", *Canadian Journal of Mathematics* 12 (1960) 145-157. MR 22 # 2558.

[12] A. Hedayat and S. Kageyama, "The Family of t-Designs - Part I", *Journal of Statistical Planning and Inference* 4 (1980) 173-212.

[13] X. Hubaut, "Two new families of 4-designs", *Discrete Mathematics* 9 (1974) 247-249. MR 49 # 10570.

[14] D.R. Hughes, "On t-designs and groups", *American Journal of Mathematics* 87 (1965) 761-778. MR 32 # 5727.

[15] W. Kantor, "4-homogeneous groups", *Mathematische Zeitschrift* 103 (1968) 67-68.

[16] E.S. Kramer, "Some t-designs for $t \geq 4$ and $v = 17, 18$", Proceedings of the Sixth Southeastern Conference on Combinatorics, Graph Theory and Computing (Boca Raton 1975). Congressus Numerantium XIV (Utilitas Math. Publ. Winnipeg, 1975) 443-460.

[17] E.S. Kramer, "The t-designs using $M_{21} \cong PSL_3(4)$ on 21 points", *Ars Combinatoria* 17 (1984) 191-208.

[18] E.S. Kramer and D.M. Mesner, "t-designs on hypergraphs", *Discrete Mathematics* 15 (1976) 263-296.

[19] E.S. Kramer, S.S. Magliveras and D.M. Mesner, "t-designs from the large Mathieu groups", *Discrete Mathematics* 36 (1981) 171-189.

[20] R.N. Lane, "t-designs and t-ply homogeneous groups", *Journal of Combinatorial Theory* (A) 10 (1971) 106-118. MR 43 # 7346.

[21] C.C. Lindner and A. Rosa, *Topics on Steiner Systems, Annals of Discrete Mathematics* 7, North-Holland Publ. Co. Amsterdam, N.Y., Oxford (1981).

[22] D. Livingstone and A. Wagner, "Transitivity of finite permutation groups on unordered sets", *Mathematische Zeitschrift* 90 (1965) 393-403.

[23] H. Luneburg, ***Transitive Erweiterungen endlicher Permutationsgruppen***, Lecture Notes in Mathematics, Springer Verlag Vol. 84 (1969), Berlin. MR 39 # 2858.

[24] S.S. Magliveras and D.W. Leavitt, "Simple 6-(33,8,36) Designs from $P\Gamma L_2(32)$", to appear in the Proceedings of the 1982 London Mathematical Soc. Symposium on Computational Group Theory, in Durham England, Acad. Press.

[25] N.S. Mendelsohn, "Intersection numbers of t-designs", In: Mirsky, ed. Studies in Pure Mathematics (presented to Richard Rado), 145-150 Academic Press, London. MR. 42 # 5819.

[26] W.H. Mills, "A new 5-design", *Ars Combinatoria* 6 (1978) 193-195.

[27] W.H. Mills, "The construction of BIBD using nonabelian groups", Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory and Computing (Boca Raton, Florida) (1978) 519-526. MR 80 # 05031.

[28] H. Nagao, *Groups and Designs*, Iwanami Shoten (1974) Tokyo.

[29] V. Pless, "On a new family of symmetry codes and related new five-designs", *Bulletin of the American Mathematical Society* 75 (1969) 1339-1342. MR 39 # 6763.

[30] V. Pless, "On a family of symmetry codes over GF(3) and related new five-designs", *Combinatorial Structures and their Applications*, (Proc. Calgary Intern. Conf., Calgary, Alberta), (1969) 323-326. Gordon and Breach, New York. MR 41 # 8263.

[31] V. Pless, "The weight of the symmetry code for $p = 29$ and the 5-designs contain therein", *Annals of the New York Academy of Sciences* 175 (1970), Art. 1 310-313.

[32] V. Pless, "Symmetry codes over GF(3) and new five-designs", *Journal of Combinatorial Theory* (A) 12 (1972) 119-142. MR 44 # 8044.

[33] D.K. Ray-Chaudhuri and R.M. Wilson, "Solution of Kirkman's Schoolgirl problem", Combinatorics (Proc. Symp. Pure Math., Vol. XIX, Univ. of Calif., Los Angeles, 1968) 187-203. AMS, Providence, R.I. MR 47 # 3195.

[34] D.K. Ray-Chaudhuri and R.M. Wilson, "On t-designs", *Osaka Journal of Mathematics* 12 (1975) 737-744. MR 52 # 13441.

[35] D.A. Sprott, "Some series of balanced incomplete block designs", *Sankhya* 17 (1956) 185-192.

[36] R.G. Stanton and D.A. Sprott, "Block intersections in balanced incomplete block designs", *Canadian Mathematical Bulletin* 7 (1964) 539-548. MR 30 # 4692.

[37] R.G. Stanton and R.C. Mullin, "Uniqueness theorems in balanced incomplete block designs", *Journal of Combinatorial Theory* 7 (1969) 37-48. MR 39 # 81.

[38] R.M. Wilson, "An existence theory for pairwise balanced designs, I: Composition theory and morphisms", *Journal of Combinatorial Theory* (A) 13 (1972) 220-245. MR 46 # 3338.

[39] R.M. Wilson, "Cyclotomy and difference families in elementary abelian groups", *Journal of Number Theory* 4 (1972) 17-47. MR 46 # 8860.

[40] R.M. Wilson, "An existence theory for pairwise balanced designs, II: The structure of PBD-closed sets and the existence conjectures", *Journal of Combinatorial Theory* (A) 13 (1972) 246-273. MR 46 # 3339.

[41] R.M. Wilson, "The necessary conditions for t-designs are sufficient for something", *Utilitas Mathematica* 4 (1973) 207-215.

[42] R.M. Wilson, "An existence theory for pairwise balanced designs, III: Proof of the existence conjectures", *Journal of Combinatorial Theory* (A) 18 (1975) 71-79. MR 51 # 2942.

[43] E. Witt, "Über Steinersche systeme", *Abh. Math. Sem. Hamb.* 12 (1938) 265-275.

Table 1

Orbit Lengths and representatives under $PSL_2(19)$, and $PGL_2(19)$, for $3 \le k \le 10$.

k	ORBIT LABEL	ORBIT REPRESENTATIVE	ORBIT LENGTH
3	1	1-3	1140
4	1-	1-3 7	285
4	1+	1-3 16	285
4	2	1-3 8	855
4	3	1-3 4	1710
4	4	1-3 5	1710
5	1	1-3 4 10	684
5	2	1-3 5 12	1140
5	3	1-3 4 5	3420
5	4	1-3 4 12	3420
5	5-	1-3 4 7	3420
5	5+	1-3 4 11	3420
6	1	1-4 12 20	570
6	2-	1-4 7 8	570
6	2+	1-4 11 15	570
6	3	1-4 5 6	1140
6	4	1-4 5 20	1710
6	5	1-4 7 10	1710
6	6	1-4 7 17	1710
6	7-	1-4 5 7	1710
6	7+	1-4 5 18	1710
6	8-	1-4 5 9	1710
6	8+	1-4 5 16	1710
6	9-	1-4 7 9	1710
6	9+	1-4 11 17	1710
6	10	1-4 5 8	3420
6	11	1-4 7 11	3420
6	12-	1-4 5 10	3420
6	12+	1-4 5 11	3420
6	13-	1-4 5 12	3420
6	13+	1-4 5 13	3420
7	1	1-4 5 10 11	1140
7	2-	1-4 5 6 9	1140
7	2+	1-4 5 6 17	1140
7	3-	1-4 7 8 9	1140
7	3+	1-4 7 11 20	1140
7	4	1-4 5 6 7	3420
7	5	1-4 5 6 10	3420
7	6	1-4 5 6 12	3420
7	7	1-4 5 7 16	3420
7	8	1-4 5 8 14	3420
7	9	1-4 5 8 15	3420
7	10	1-4 5 9 12	3420
7	11	1-4 5 9 14	3420
7	12	1-4 5 10 13	3420
7	13-	1-4 5 7 8	3420
7	13+	1-4 5 8 11	3420
7	14-	1-4 5 7 9	3420
7	14+	1-4 5 11 12	3420
7	15-	1-4 5 7 11	3420
7	15+	1-4 5 8 10	3420
7	16-	1-4 5 7 12	3420
7	[illegible]	1-4 5 8 18	3420
7	17-	1-4 5 7 13	3420
7	17+	1-4 5 12 18	3420
7	18-	1-4 5 8 9	3420
7	18+	1-4 5 8 13	3420
8	1	1-5 6 9 17	285
8	2	1-5 9 14 15	570
8	3-	1-5 7 13 20	570
8	3+	1-5 12 18 20	570
8	4	1-5 8 9 13	855
8	5-	1-5 7 8 10	855
8	5+	1-5 7 11 15	855
8	6	1-5 6 7 8	1710
8	7	1-5 7 12 20	1710
8	8	1-5 7 13 16	1710
8	9-	1-5 6 10 12	1710
8	9+	1-5 6 10 20	1710
8	10-	1-5 7 8 14	1710
8	10+	1-5 8 11 14	1710
8	11-	1-5 7 9 14	1710
8	11+	1-5 8 9 14	1710
8	12-	1-5 7 12 17	1710
8	12+	1-5 8 13 18	1710
8	13-	1-5 7 12 18	1710
8	13+	1-5 7 13 18	1710
8	14	1-5 6 7 12	3420
8	15	1-5 6 7 20	3420
8	16	1-5 6 10 13	3420
8	17	1-5 6 10 14	3420
8	18	1-5 6 12 14	3420
8	19	1-5 7 8 11	3420
8	20	1-5 7 11 17	3420
8	21-	1-5 6 7 9	3420
8	21+	1-5 6 7 11	3420
8	22-	1-5 6 7 10	3420
8	22+	1-5 6 7 17	3420
8	23-	1-5 6 7 13	3420
8	23+	1-5 6 7 14	3420
8	24-	1-5 6 9 10	3420
8	24+	1-5 6 10 17	3420
8	25-	1-5 6 9 12	3420
8	25+	1-5 6 12 17	3420
8	26-	1-5 7 8 9	3420
8	26+	1-5 8 11 13	3420
8	27-	1-5 7 8 12	3420
8	27+	1-5 8 11 18	3420
8	28-	1-5 7 8 15	3420
8	28+	1-5 8 10 11	3420
8	29-	1-5 7 8 18	3420
8	29+	1-5 7 12 16	3420
8	30-	1-5 7 9 13	3420
8	30+	1-5 7 11 12	3420
8	31-	1-5 7 9 16	3420
8	31+	1-5 7 11 18	3420
9	1	1-5 6 10 13 16	380
9	2	1-5 6 12 14 20	1140
9	3-	1-5 6 7 9 16	1140
9	3+	1-5 6 7 11 18	1140
9	4	1-5 6 7 8 9	3420
9	5	1-5 6 7 8 12	3420
9	6	1-5 6 7 8 14	3420
9	7	1-5 6 7 9 11	3420
9	8	1-5 6 7 9 17	3420
9	9	1-5 6 7 10 15	3420
9	10	1-5 6 7 10 17	3420
9	11	1-5 6 7 12 13	3420
9	12	1-5 6 7 12 15	3420
9	13	1-5 6 9 10 20	3420
9	14	1-5 7 8 9 14	3420
9	15	1-5 7 8 12 18	3420
9	16-	1-5 6 7 8 10	3420
9	16+	1-5 6 7 8 13	3420
9	17-	1-5 6 7 9 10	3420
9	17+	1-5 6 7 11 17	3420
9	18-	1-5 6 7 9 13	3420
9	18+	1-5 6 7 11 12	3420
9	19-	1-5 6 7 9 14	3420
9	19+	1-5 6 7 10 11	3420
9	20-	1-5 6 7 9 15	3420
9	20+	1-5 6 7 11 20	3420
9	21-	1-5 6 7 9 20	3420
9	21+	1-5 6 7 11 14	3420
9	22-	1-5 6 7 10 13	3420
9	22+	1-5 6 7 14 17	3420
9	23-	1-5 6 7 10 14	3420
9	23+	1-5 6 7 13 17	3420
9	24-	1-5 6 7 10 20	3420
9	24+	1-5 6 7 17 20	3420
9	25-	1-5 6 7 13 20	3420
9	25+	1-5 6 7 14 20	3420
9	26-	1-5 6 9 10 12	3420
9	26+	1-5 6 10 14 20	3420
9	27-	1-5 6 9 10 13	3420
9	27+	1-5 6 10 13 17	3420
9	28-	1-5 6 9 10 14	3420
9	28+	1-5 6 10 14 17	3420
9	29-	1-5 6 9 12 14	3420
9	29+	1-5 6 12 14 17	3420
9	30-	1-5 7 8 9 10	3420
9	30+	1-5 7 8 11 15	3420
9	31-	1-5 7 8 9 15	3420
9	31+	1-5 7 11 17 18	3420
9	32-	1-5 7 8 11 13	3420
9	32+	1-5 7 8 11 18	3420
9	33-	1-5 7 9 13 16	3420
9	33+	1-5 7 11 12 18	3420
10	1	1-5 6 7 12 15 20	342
10	2-	1-5 7 9 13 16 20	342
10	2+	1-5 7 11 12 18 20	342
10	3-	1-5 6 9 10 13 16	380
10	3+	1-5 6 10 13 16 17	380
10	4-	1-5 6 7 9 10 16	1140
10	4+	1-5 6 7 11 17 18	1140
10	5-	1-5 6 7 9 11 13	1140
10	5+	1-5 6 7 9 11 18	1140
10	6-	1-5 6 9 12 14 20	1140
10	6+	1-5 6 12 14 17 20	1140
10	7	1-5 6 7 8 10 13	1710
10	8	1-5 6 7 8 14 20	1710
10	9	1-5 6 7 9 15 17	1710
10	10	1-5 6 7 10 17 20	1710
10	11	1-5 7 8 11 12 18	1710
10	12-	1-5 6 7 8 9 11	1710
10	12+	1-5 6 7 8 9 18	1710
10	13-	1-5 6 7 8 10 15	1710
10	13+	1-5 6 7 8 13 18	1710
10	14-	1-5 6 7 8 10 20	1710
10	14+	1-5 6 7 8 13 14	1710
10	15-	1-5 6 7 9 10 17	1710
10	15+	1-5 6 7 9 14 17	1710
10	16-	1-5 6 7 10 13 17	1710
10	16+	1-5 6 7 10 14 17	1710
10	17-	1-5 6 7 10 13 20	1710
10	17+	1-5 6 7 14 17 20	1710
10	18-	1-5 6 9 10 12 13	1710
10	18+	1-5 6 9 10 14 20	1710
10	19-	1-5 6 9 10 12 14	1710
10	19+	1-5 6 10 14 17 20	1710
10	20-	1-5 7 8 9 10 14	1710
10	20+	1-5 7 8 9 10 15	1710
10	21	1-5 6 7 8 9 10	3420
10	22	1-5 6 7 8 9 12	3420
10	23	1-5 6 7 8 9 20	3420
10	24	1-5 6 7 8 10 18	3420
10	25	1-5 6 7 9 10 20	3420
10	26	1-5 6 7 9 11 14	3420
10	27	1-5 6 7 9 13 20	3420
10	28	1-5 6 7 10 14 15	3420
10	29-	1-5 6 7 8 9 13	3420
10	29+	1-5 6 7 8 9 16	3420
10	30-	1-5 6 7 8 9 14	3420
10	30+	1-5 6 7 8 9 15	3420
10	31-	1-5 6 7 8 10 12	3420
10	31+	1-5 6 7 8 12 13	3420
10	32-	1-5 6 7 8 10 14	3420
10	32+	1-5 6 7 8 13 20	3420
10	33-	1-5 6 7 8 10 16	3420
10	33+	1-5 6 7 8 12 18	3420
10	34-	1-5 6 7 8 12 14	3420
10	34+	1-5 6 7 8 12 20	3420
10	35-	1-5 6 7 9 10 13	3420
10	35+	1-5 6 7 10 13 15	3420
10	36-	1-5 6 7 9 10 14	3420
10	36+	1-5 6 7 10 11 17	3420
10	37-	1-5 6 7 9 10 15	3420
10	37+	1-5 6 7 10 15 17	3420
10	38-	1-5 6 7 9 10 18	3420
10	38+	1-5 6 7 9 11 17	3420
10	39-	1-5 6 7 9 13 14	3420
10	39+	1-5 6 7 9 13 15	3420
10	40-	1-5 6 7 9 13 16	3420
10	40+	1-5 6 7 11 12 18	3420
10	41-	1-5 6 7 9 14 15	3420
10	41+	1-5 6 7 10 11 20	3420
10	42-	1-5 6 7 9 14 20	3420
10	42+	1-5 6 7 10 11 14	3420
10	43-	1-5 6 7 9 15 20	3420
10	43+	1-5 6 7 10 15 20	3420
10	44-	1-5 6 7 10 14 20	[illegible]
10	44+	1-5 6 7 13 17 20	3420

Table 2

Matrices for $PSL_2(19)$ and implicitly for $PGL_2(19)$ for $3 \le t < k \le 10$ and $t \le 7$, but not for $t = 7$ and $k = 8$.

k \ t	3	4					5						6																		
4	1																														
	1																														
	3																														
	6																														
	6																														
5	6	0	0	0	2	0																									
	10	6	6	0	0	2																									
	30	0	0	0	6	6																									
	30	0	0	8	2	4																									
	30	12	0	4	6	2																									
	30	0	12	4	6	2																									
6	10	0	0	6	1	2	0	0	0	1	0	0																			
	10	6	0	2	2	1	0	0	0	0	1	0																			
	10	0	6	2	2	1	0	0	0	0	0	1																			
	20	0	0	0	4	6	0	0	2	0	0	0																			
	30	0	0	8	4	7	0	0	1	2	0	0																			
	30	6	6	4	8	2	5	0	0	0	1	1																			
	30	12	12	4	4	5	0	3	0	0	1	1																			
	30	6	0	6	5	6	0	0	1	1	1	0																			
	30	0	6	6	5	6	0	0	1	1	0	1																			
	30	12	6	2	4	7	0	3	1	0	1	0																			
	30	6	12	2	4	7	0	3	1	0	0	1																			
	30	12	0	8	5	4	0	0	0	1	2	0																			
	30	0	12	8	5	4	0	0	0	1	0	2																			
	60	12	12	8	12	20	0	0	2	0	2	2																			
	60	12	12	16	10	8	0	0	0	2	2	2																			
	60	12	0	8	16	10	5	0	2	1	2	0																			
	60	0	12	8	14	10	5	0	2	1	0	2																			
	60	12	12	12	8	12	0	3	1	2	1	1																			
	60	12	12	12	8	12	0	3	1	2	1	1																			
7	35	4	6	8	10	8	5	0	2	2	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
	35	12	8	6	8	10	0	1	1	0	2	1	0	0	0	2	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
	35	8	12	6	8	10	0	1	3	0	1	2	0	0	0	1	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0
	35	16	6	12	8	6	0	0	0	2	4	1	0	2	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0
	35	6	16	12	8	6	0	0	0	2	1	4	0	0	2	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0
	105	12	12	16	28	30	3	0	11	3	3	3	0	0	0	6	0	0	0	2	2	0	0	0	0	1	0	1	1	0	0
	105	24	24	16	26	30	5	9	8	3	3	3	0	0	0	1	0	0	0	0	0	2	2	0	0	0	0	1	1	1	1
	105	12	12	26	24	30	5	4	8	7	2	2	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	105	24	24	24	10	30	0	9	6	6	3	3	0	0	0	0	2	0	0	2	2	1	2	0	0	0	0	0	0	1	1
	105	30	36	26	32	18	15	0	3	3	6	6	0	0	0	0	0	4	0	0	0	0	0	2	2	1	0	1	1	0	0
	105	24	24	24	26	26	9	3	5	4	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	1	1	1	1
	105	36	36	26	20	26	0	12	3	4	5	5	0	0	0	0	0	0	2	0	0	2	2	0	0	0	2	0	0	1	1
	105	36	36	16	26	56	10	12	4	1	3	5	0	0	0	0	0	2	6	0	0	2	2	0	0	0	0	1	1	0	0
	105	26	46	46	10	20	15	1	3	4	5	5	0	0	0	0	0	6	0	0	0	0	0	0	0	0	1	1	1	1	1
	105	36	12	24	30	20	10	0	4	3	9	3	0	6	0	0	0	2	0	2	0	0	0	0	0	1	1	2	0	0	0
	105	12	36	24	30	20	10	0	4	1	3	9	0	0	6	0	0	2	0	0	2	0	0	0	0	1	1	0	2	0	0
	105	36	12	26	26	26	3	6	3	3	7	3	0	0	0	0	0	0	0	2	0	2	0	2	0	0	0	2	0	1	1
	105	12	36	26	26	16	5	6	5	5	3	7	0	0	0	0	0	0	0	0	2	0	2	0	2	0	0	0	2	1	1
	105	32	24	28	26	36	3	0	5	6	3	6	0	0	0	0	2	0	0	2	0	0	0	0	2	1	1	0	2	0	0
	105	24	12	28	26	26	5	0	5	6	6	3	0	0	0	0	2	0	0	0	2	0	0	2	0	1	1	2	0	0	0
	105	36	24	28	22	24	0	6	3	3	7	6	0	0	0	0	0	0	2	1	0	0	0	2	0	1	1	0	0	2	0
	105	26	36	28	22	26	0	6	3	5	4	7	0	0	0	0	0	0	2	0	2	0	0	0	2	1	1	0	0	0	2
	105	26	32	36	20	26	0	3	3	10	5	2	6	0	0	0	2	0	0	2	0	0	0	2	0	0	1	0	0	0	2
	105	12	26	36	20	26	0	3	3	10	2	5	6	0	0	0	2	0	0	0	2	0	0	0	2	0	1	0	0	1	0
	105	48	26	26	22	26	0	9	3	3	9	3	0	6	0	0	0	0	2	0	0	2	0	2	0	2	0	0	0	1	1
	105	26	48	36	22	24	0	9	3	3	3	9	0	0	6	0	0	0	2	0	0	0	2	0	2	1	0	0	0	1	1
8	16	5	5	2	6	5	0	2	2	0	1	1	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
	28	10	10	6	9	8	3	6	2	2	2	2	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0
	28	10	4	12	7	8	0	1	1	6	1	1	3	1	0	0	1	0	0	1	0	0	0	2	0	0	1	0	0	0	1
	28	4	10	12	7	8	0	1	1	6	1	1	1	0	1	0	1	0	0	0	1	0	0	0	2	0	1	0	0	1	0
	42	18	18	12	11	12	0	6	2	2	6	6	0	3	3	0	0	0	2	0	0	1	1	1	1	1	0	0	0	1	1
	42	15	6	13	16	11	5	0	3	3	3	2	0	3	0	0	1	1	0	1	1	0	0	1	0	1	1	2	0	0	0
	42	6	15	13	16	11	5	0	3	3	2	5	0	0	3	0	1	1	0	1	1	0	0	0	1	1	1	0	2	0	0
	84	18	18	20	29	25	10	0	10	6	6	6	0	3	3	6	0	2	0	3	3	0	0	0	0	2	1	2	2	0	0
	84	26	26	32	21	25	0	6	6	10	6	6	6	0	0	0	2	0	2	2	2	0	0	2	2	1	2	0	0	2	2
	84	24	24	30	20	25	0	9	5	10	5	5	6	0	0	0	3	0	1	2	1	2	1	1	1	0	2	0	0	2	2
	84	26	18	22	21	29	5	9	9	7	3	3	0	0	0	1	3	0	0	2	1	1	2	1	0	0	0	2	1	2	2
	84	18	26	22	21	29	5	9	9	7	3	3	0	0	0	1	3	0	0	1	2	2	1	0	1	0	0	1	2	2	2
	84	30	18	26	29	21	15	3	5	3	9	5	0	3	0	0	0	9	0	3	0	1	0	2	3	1	5	3	1	1	1
	84	18	30	26	29	21	15	3	5	3	5	9	0	0	1	0	0	5	0	0	2	0	1	1	2	1	1	1	3	1	1
	84	30	30	22	26	23	10	9	5	6	7	7	0	0	1	0	0	3	1	1	0	2	2	2	2	1	0	2	1	1	1
	84	30	30	22	26	23	10	9	8	6	7	7	0	1	0	0	0	3	1	0	1	2	2	1	2	1	0	1	3	1	1
	84	42	26	28	22	23	0	9	3	6	11	5	0	6	0	0	0	0	3	1	0	2	1	3	0	2	3	0	0	2	2
	84	26	42	28	22	23	0	9	3	6	5	11	0	0	6	0	0	0	1	0	1	1	2	0	5	1	3	0	0	1	2
	84	24	30	28	21	26	0	9	3	8	5	7	1	0	1	0	2	0	2	2	2	1	1	1	2	1	1	0	0	1	1
	84	30	24	28	21	26	0	9	5	8	7	5	1	1	0	0	2	0	2	2	2	2	1	2	1	1	1	0	0	1	1
	168	36	36	68	50	56	10	6	18	16	10	10	0	0	0	0	6	0	2	6	6	0	0	2	2	1	2	3	3	2	2
	168	36	36	52	52	50	30	6	16	16	10	10	6	0	0	6	6	6	0	2	2	0	0	2	1	1	2	3	3	3	3
	168	60	60	40	50	50	30	24	12	8	12	12	0	0	0	3	0	6	6	0	0	6	6	0	0	0	3	3	3	3	3
	168	48	48	32	48	50	30	12	12	16	12	12	6	6	6	3	6	2	2	2	2	3	2	2	2	2	3	2	2	3	3
	168	36	36	56	68	52	10	6	16	18	10	10	6	0	0	3	0	0	0	2	2	0	0	2	2	3	3	3	3	3	3
	168	48	68	56	56	60	20	0	8	12	16	16	0	6	6	0	2	6	0	2	2	0	0	6	6	3	3	3	3	0	0
	168	48	68	52	36	66	20	6	10	12	16	16	0	0	0	0	2	6	2	2	2	0	0	6	4	6	3	3	3	2	2
	168	68	26	40	50	56	10	12	20	10	14	8	0	0	0	0	2	0	0	6	6	6	2	2	0	6	3	6	2	2	2
	168	36	60	60	50	56	10	12	20	10	8	12	0	0	0	0	3	0	0	6	6	2	6	0	2	6	1	2	6	3	2
	168	60	48	60	52	30	15	15	15	7	15	11	0	6	0	6	0	6	6	2	2	6	2	2	2	6	1	3	3	1	2
	168	48	60	40	52	30	15	15	15	7	11	15	0	0	6	6	0	6	6	2	2	2	6	2	2	6	1	2	3	2	1
	168	36	36	68	50	56	15	9	17	15	8	9	6	0	0	6	6	0	0	6	2	2	2	2	2	3	3	4	5	2	3
	168	36	36	68	50	56	15	9	17	15	9	9	6	0	0	6	6	0	0	2	4	2	2	2	2	1	1	5	6	3	3
	168	72	68	46	66	52	5	21	13	9	17	9	0	6	0	1	0	0	6	2	0	8	2	6	0	6	2	2	1	4	1
	168	68	72	64	66	52	5	21	13	9	9	17	0	0	6	3	0	0	6	0	2	2	8	0	4	6	2	1	2	1	6
	168	68	60	66	68	52	10	18	16	10	10	16	0	0	6	3	6	2	6	2	2	6	6	0	2	2	2	1	3	2	2
	168	60	66	66	68	52	10	18	16	10	16	10	0	6	0	1	6	2	6	2	2	6	6	2	0	1	2	3	1	2	2
	168	72	36	52	52	66	15	9	9	21	21	9	0	15	0	0	0	6	2	4	0	2	0	6	0	3	6	6	1	3	3
	168	36	72	52	52	66	15	9	9	21	9	21	0	0	12	0	0	4	2	0	6	0	2	0	6	3	4	1	6	2	3
	168	72	68	48	48	68	10	18	10	10	18	10	0	6	0	0	0	2	6	4	0	6	2	6	0	3	3	3	1	6	2
	168	68	72	68	68	68	10	18	10	10	10	18	0	0	6	0	0	2	6	0	6	2	4	0	4	3	3	1	3	2	6
	168	68	36	52	56	66	25	3	11	13	15	11	6	6	0	0	2	6	0	6	0	0	0	2	2	1	1	5	4	1	2
	168	24	68	32	36	46	25	3	13	13	11	15	6	0	6	0	2	6	0	0	4	0	0	1	2	3	1	4	5	2	1
	168	60	68	48	50	68	15	15	11	11	15	11	0	6	0	0	2	6	4	6	2	4	2	3	2	2	2	3	2	2	1
	168	68	60	68	50	68	15	15	11	11	11	15	0	0	6	0	2	4	6	2	4	2	6	2	2	2	2	2	3	1	2
	168	60	36	60	66	68	5	9	9	17	17	9	6	6	0	0	6	0	2	2	4	2	0	3	2	2	6	3	0	2	3
	168	36	60	60	66	68	5	8	9	17	9	17	6	6	6	0	6	0	2	4	2	0	2	2	8	2	6	0	3	1	2
	168	60	36	76	44	52	5	15	11	17	15	7	6	0	0	0	4	0	3	6	2	6	2	6	0	1	1	1	0	1	6
	168	36	60	56	64	52	5	15	11	17	7	15	6	0	0	0	4	0	2	2	6	2	4	0	6	1	1	0	2	6	1

Table 2 (cont.)

Matrices for $PSL_2(19)$ and implicitly for $PGL_2(19)$ for $3 \leq t < k \leq 10$ and $t \leq 7$, but not for $t = 7$ and $k = 8$.

k \ t	3	4					5						6	7
9	28	12	12	8	10	10	3	6	3	2	3	3	[illegible]	[illegible]
	84	24	14	32	32	28	13	3	9	11	9	9	[illegible]	[illegible]
	84	36	14	28	28	32	3	12	11	9	11	6	[illegible]	[illegible]
	84	24	36	28	28	12	3	12	11	9	6	11	[illegible]	[illegible]
	232	84	84	80	96	88	33	18	34	21	29	29	[illegible]	[illegible]
	232	96	84	80	90	94	23	27	36	24	26	26	[illegible]	[illegible]
	232	72	72	88	98	86	30	12	31	29	26	26	[illegible]	[illegible]
	232	72	72	80	88	100	23	27	41	29	21	21	[illegible]	[illegible]
	232	96	96	72	88	96	20	39	37	18	27	27	[illegible]	[illegible]
	232	96	96	88	90	86	23	27	26	24	31	31	[illegible]	[illegible]
	232	108	108	80	88	88	23	42	26	19	31	31	[illegible]	[illegible]
	232	72	72	88	86	98	23	27	36	34	21	21	[illegible]	[illegible]
	232	72	72	96	90	90	30	13	30	33	23	23	[illegible]	[illegible]
	232	96	96	80	86	94	23	42	11	24	26	26	[illegible]	[illegible]
	232	96	96	88	96	80	43	24	22	23	32	32	[illegible]	[illegible]
	232	96	96	96	82	90	13	36	23	32	28	28	[illegible]	[illegible]
	232	84	72	96	96	90	33	18	34	26	29	24	[illegible]	[illegible]
	232	72	84	84	94	90	33	18	34	26	24	29	[illegible]	[illegible]
	232	108	84	80	92	86	15	33	29	21	34	24	[illegible]	[illegible]
	232	84	108	80	92	88	33	33	29	21	24	34	[illegible]	[illegible]
	232	84	72	92	86	96	20	24	32	33	27	22	[illegible]	[illegible]
	232	72	96	92	86	94	20	24	32	33	22	27	[illegible]	[illegible]
	232	96	96	96	90	90	30	30	30	23	23	30	[illegible]	[illegible]
	232	96	84	84	90	90	20	30	30	23	10	23	[illegible]	[illegible]
	232	96	72	88	88	92	23	27	31	28	31	21	[illegible]	[illegible]
	232	72	96	88	88	92	23	27	31	29	21	31	[illegible]	[illegible]
	232	84	84	88	88	92	30	30	30	30	23	23	[illegible]	[illegible]
	232	84	84	88	88	92	30	30	30	30	23	23	[illegible]	[illegible]
	232	96	96	80	92	88	40	36	28	22	28	28	[illegible]	[illegible]
	232	96	96	80	92	88	40	36	28	22	28	28	[illegible]	[illegible]
	232	96	84	92	88	88	23	27	26	29	11	26	[illegible]	[illegible]
	232	84	96	92	88	88	23	27	26	29	26	31	[illegible]	[illegible]
	232	96	96	96	90	90	33	33	29	26	24	29	[illegible]	[illegible]
	232	96	84	84	90	90	33	33	29	26	29	24	[illegible]	[illegible]
	232	84	72	100	90	86	30	13	23	33	30	23	[illegible]	[illegible]
	232	72	96	100	90	86	30	13	23	33	23	30	[illegible]	[illegible]
	232	96	96	92	82	96	19	26	28	22	26	23	[illegible]	[illegible]
	232	84	96	92	82	94	13	36	28	32	23	28	[illegible]	[illegible]
	232	132	84	88	86	86	20	39	22	23	42	22	[illegible]	[illegible]
	232	84	132	88	86	86	20	39	22	23	22	42	[illegible]	[illegible]
	232	96	108	92	84	88	13	36	23	27	28	33	[illegible]	[illegible]
	232	108	96	92	84	88	13	36	23	27	33	28	[illegible]	[illegible]
	232	72	108	92	88	88	23	27	26	28	21	36	[illegible]	[illegible]
	232	108	72	92	88	88	23	27	26	29	36	21	[illegible]	[illegible]
	232	96	72	86	98	78	43	9	22	28	37	27	[illegible]	[illegible]
	232	72	96	96	98	78	43	9	22	28	27	37	[illegible]	[illegible]
	232	108	72	92	96	82	40	21	23	27	38	23	[illegible]	[illegible]
	232	72	108	92	96	82	40	21	23	27	23	38	[illegible]	[illegible]
	232	96	96	96	88	96	25	27	21	29	31	31	[illegible]	[illegible]
	232	96	96	96	88	84	23	27	21	29	11	31	[illegible]	[illegible]
	232	108	72	108	78	90	3	30	20	40	13	20	[illegible]	[illegible]
	232	72	108	108	78	90	5	30	20	40	20	33	[illegible]	[illegible]

Table 2 (cont.)

Matrices for $PSL_2(19)$ and implicitly for $PGL_2(19)$ for $3 \le t < k \le 10$ and $t \le 7$, but not for $t=7$ and $k=8$.

10

Table 3.

List of parameters t, k, λ with $3 \le t < k \le 10$ and $\lambda \le \bar{\lambda}/2$ for which there is a simple, nontrivial t-(20,k,λ) design with $PGL_2(19)$ or $PSL_2(19)$ as an automorphism group.

t	k	$\underline{\lambda}$	$\bar{\lambda}$	λ values using $PGL_2(19)$	NEW λ values using $PSL_2(19)$
3	4	1	17	2, 3, 5, 6, 8	1, 4, 7
3	5	2	136	0, 6, 10, 16 (mod 30)	NONE
4	5	4	16	NONE	NONE
3	6	10	680	10x, $1 \le x \le 34$	NONE
4	6	30	120	NONE	60
5	6	15	15	NONE	NONE
3	7	35	2380	35x, $1 \le x \le 34$	NONE
4	7	140	560	280	NONE
5	7	105	105	NONE	NONE
6	7	14	14	NONE	NONE
3	8	14	6188	14x, $1 \le x \le 221$	NONE
4	8	70	1820	70x, $1 \le x \le 13$ but <u>not</u> 140,490	140, 490
5	8	35	455	35x, $2 \le x \le 6$	35
6	8	7	91	NONE	NONE
7	8	1	13	NONE	NONE
3	9	28	12376	28x, $1 \le x \le 221$, $x \not\equiv 2 \pmod 3$	NONE
4	9	168	4368	168x, $2 \le x \le 12$, $x \not\equiv 1 \pmod 3$	NONE
5	9	105	1365	105x, $1 \le x \le 6$, $x \not\equiv 2 \pmod 3$	NONE
6	9	28	364	NONE	112 (2 nonisomorphic solutions)
7	9	6	78	NONE	NONE
8	9	12	12	NONE	NONE
3	10	4	19448	40x+36y, $x \not\equiv 1 \pmod 3$, $x \ne 3$ or 5, and $y \in \{0,1,2,3,5,6,7,8\}$	40x+36y, $x \equiv 1 \pmod 3$ or $x = 3$ or 5, $y \in \{0,1,2,3,5,6,7,8\}$
4	10	28	8008	28x, $21 \le x \le 142$, $\lambda \equiv 0$, or $4 \pmod 6$ and $\lambda \not\equiv 4 \pmod 5$ and $\lambda \ne 616, 868$	28x, $\lambda \equiv 2 \pmod 6$, $x \ge 11$, $\lambda \not\equiv 4 \pmod 5$, also get 252, 336, 420, 448, 532, 616, 868.
5	10	21	3003	21x, $12 \le x \le 71$, $\lambda \not\equiv 4 \pmod 5$ and $x \not\equiv 1 \pmod 3$	21x, $x \ge 7$, $x \equiv 1 \pmod 3$ and $\lambda \not\equiv 4 \pmod 5$; get 126, 168, 231
6	10	7	1001	NONE	
7	10	2	286	NONE	
8	10	6	66	NONE	
9	10	1	11	NONE	NONE

Table 4.

λ_4	λ_5	Orbits for 4-(20,8,λ_4) and/or 5-(20,8,λ_5) designs.
70		2 4 6 7
140	35	2 5- 6 11+ 15- 17
210		4 6 8 9 26
280	70	1 3 4 5 6 9 11 12 15
350		1 3 8 15 16 22 30
420	105	1 2 4 5 6 7 8 15 16 22 30
490		1 2 5- 5 6 8 11+ 13- 15 16 17 22 30
560	140	3 6 7 12 13 15 16 21 27 28
630		2 3 4 6 7 13 21 22 27 28 30
700	275	2 6 8 10 11 12 13 14 16 17 19 20 21 31
770		3 4 6 8 9 13 21 22 26 27 28 30 .
840	210	6 7 10 11 13 15 16 17 21 22 26 30 31
910		2 4 6 7 19 21 22 23 26 27 29 30 31

λ_4	λ_5	Orbits for 4-(20,9,λ_4) and/or 5-(20,9,λ_5) designs.
336	105	1 3 6 9 15
504		3 4 23 29
840		1 2 5 16 23 28 29
1008	315	2 4 6 13 16 22 28 33
1344	420	1 4 5 6 12 14 16 22 26 27 33
1512		4 26 17 19 20 23 24 32 33
1848		1 3 4 6 9 15 16 17 19 20 23 24 32 33
2016	630	3 4 5 6 12 14 15 16 21 23 24 26 27 29 32

λ_4	Orbits producing some 4-(20,10,λ_4) designs.
252	1 2+ 17- 29+
308	1 1 3+ 5- 6+ 13+ 15+ OR 1 2 3+ 5- 6+ 15+ 17-
336	2+ 5 20- 37-
392	1 2- 5- 5- 16- 20- 26 OR 2 3- 5- 8 9 15+ 17-
420	5 6 15- 16+ 18+
448	1 2 3 8 15- 18+ 34-
476	2+ 5- 4 6+ 14+ 15+ 39- OR 1 3- 4 5- 14+ 17 19-
532	1 2- 3 5- 6- 8 14- 15- 18
560	3+ 4+ 6+ 14+ 17- 18- 19+ 34+
588	1 2 6 9 11 12 15
616	2+ 3 4 5 10 13+ 16+ 32-
672	2 4 6 18 34
700	3 7 8 10 14 16 27 OR 3 7 8 9 14 16 28
728	1 2 3- 4 5- 8 9 14- 26 36-
756	1 10 11 13 17 24 26
812	1 2+ 3- 4- 5+ 17- 33+ 34+ 37+ 42+
840	4 9 10 12 24 42 OR 4 26 41 43 OR 5 12 37 42
840	5 15 20 21 40 OR 6 12 16 27 41 OR 6 14 16 28 29
868	1 2 3 5 6 7 10 12- 15- 17- 18- 19- 20-
952	2 3 15 17 20 27 31
980	3- 5- 6 7 11 12 14- 15 18+ 19 20+
1008	1 2 11 12 13 18 21 29
1036	1 3 5 9 11 21 33 35
1092	2 5 9 13 20 31 36
1120	3 5 6 13 16 17 19 34
1176	1 5 6 11 13 26 33 37
1260	7 8 9 21 22 25 40 43 OR 10 13 23 33 37 39
1260	9 17 23 25 27 34 44 OR 10 12 13 21 22 24 37 44
1260	10 12 17 21 22 24 27 44
1372	2 3 5 6 9 13 17 27 34 37
1512	2 14 16 17 23 33 37 39
1540	3 5 7 8 10 13 17 31 35 44
1680	5 6 13 15 28 27 31 35 44
1792	2 3 6 13 14 27 31 33 35 37
1960	3 13 16 17 23 27 33 34 37 39
2016	1 11 12 18 21 22 24 29 30 40 42
2100	5 10 13 16 17 20 23 27 28 31 33 37
2380	3 5 6 9 13 16 23 27 33 34 35 37 39
2520	21 22 25 29 30 33 36 39 40 43
2520	21 22 28 29 30 35 38 39 43 44

λ_5	Orbits producing some 5-(20,10,λ_5) designs.
126	2+ 4- 6+ 18+ 34+
147	1 2+ 3- 5- 7 8 26+ 18+ OR 1 2- 3+ 4- 7 13+ 14- 19-
168	1 2 3 8 15- 18+ 34-
210	3- 4- 5+ 11 12- 13+ 14+ 16- 17- OR 3- 5+ 6- 29- 41+ 42+
231	2+ 3 4 5 10 13+ 16+ 32-
252	2 5 6 13 16 26
273	1 2 3+ 4- 6 16- 34- 38- 43+ OR 1 2 3+ 4- 5 8 9 13+ 14+ 16 17-
315	4 9 10 34 42 OR 5 9 10 30 44
336	1 3+ 5+ 19- 26 29+ 30+ 36+ 43+ OR 1 3- 4+ 7 11 12 13- 14- 15- 17 19
357	2 3 7 8 12 24 25 44
378	1 2 7 8 11 12 14 18 37 OR 1 2 7 8 11 15 18 20 31
420	3 5 6 21 24 41 44
441	1 5 6 11 13 26 33 37
483	1 2 3 5 11 12 13 16 17 21 25 27
525	3- 4+ 9 12- 21 26 32- 36+ 37- 39+ 40+ 41-
546	1 3 7 8 11 14 21 31 37 44
567	2 7 8 14 16 21 22 25 39 44
630	5 6 19 33 36 41 43
735	3 7 8 15 17 20 21 31 37 39 40
756	1 7 8 11 12 14 16 21 24 25 37 39 44
840	3+ 5+ 6 21 23 28 29- 31- 34- 35 36+ 38+ 39+ 41+ 44
945	22 23 24 25 27 29- 31- 32+ 33 35- 37+ 41+ 43 44
1050	3 6 15 17 21 22 31 33 35 39 41 43 44

Table 5.

Number x_i of blocks in B with a specified intersection size with a fixed orbit representative.

Orbit	Orbit Representative	Orbit Length	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	1-5 6 10 13 16	380	10	495	3582	11970	18018	12600	4410	558	36	1
2	1-5 6 12 14 20	1140	19	444	3687	11907	17913	12831	4221	633	24	1
3-	1-5 6 7 9 16	1140	16	459	3666	11886	18018	12684	4326	594	30	1
3+	1-5 6 7 11 18	1140	16	462	3645	11949	17913	12789	4263	615	27	1
5	1-5 6 7 8 12	3420	19	442	3701	11865	17983	12761	4263	619	26	1
16-	1-5 6 7 8 10	3420	13	479	3610	11970	17948	12712	4326	590	31	1
17-	1-5 6 7 9 10	3420	16	461	3652	11928	17948	12754	4284	608	28	1
18+	1-5 6 7 11 12	3420	18	446	3701	11837	18053	12677	4319	599	29	1
21-	1-5 6 7 9 20	3420	17	455	3666	11914	17948	12768	4270	614	27	1
23-	1-5 6 7 10 14	3420	18	448	3687	11879	17983	12747	4277	613	27	1
23+	1-5 6 7 13 17	3420	18	447	3694	11858	18018	12712	4298	606	28	1
24-	1-5 6 7 10 20	3420	15	466	3645	11921	17983	12705	4319	595	30	1
24+	1-5 6 7 17 20	3420	18	448	3687	11879	17983	12747	4277	613	27	1
26+	1-5 6 10 14 20	3420	18	447	3694	11858	18018	12712	4298	606	28	1
28+	1-5 6 9 10 14	3420	19	441	3708	11844	18018	12726	4284	612	27	1
30+	1-5 7 8 11 15	3420	17	455	3666	11914	17948	12768	4270	614	27	1
32-	1-5 7 8 11 12	3420	18	447	3694	11858	18018	12712	4298	606	28	1
32+	1-5 7 8 11 18	3420	18	448	3687	11879	17983	12747	4277	613	27	1

Orbit	Orbit Representative	Orbit Length	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	1-5 6 10 13 16	380	10	495	3582	11970	18018	12600	4410	558	36	1
2	1-5 6 12 14 20	1140	13	483	3582	12054	17808	12852	4242	618	27	1
3-	1-5 6 7 9 16	1140	19	441	3708	11844	18018	12726	4284	612	27	1
3+	1-5 6 7 11 18	1140	16	462	3645	11949	17913	12789	4263	615	27	1
5	1-5 6 7 8 12	3420	20	436	3715	11851	17983	12775	4249	625	25	1
7	1-5 6 7 9 11	3420	15	465	3652	11900	18018	12670	4340	588	31	1
15	1-5 7 8 12 18	3420	20	434	3729	11809	18053	12705	4291	611	27	1
17-	1-5 6 7 9 10	3420	17	455	3666	11914	17948	12768	4270	614	27	1
17+	1-5 6 7 11 17	3420	17	455	3666	11914	17948	12768	4270	614	27	1
20+	1-5 6 7 11 20	3420	15	467	3638	11942	17948	12740	4298	602	29	1
21-	1-5 6 7 9 20	3420	16	461	3652	11928	17948	12754	4248	608	28	1
22-	1-5 6 7 10 13	3420	16	460	3659	11907	17983	12719	4305	601	29	1
25-	1-5 6 7 13 20	3420	16	460	3659	11907	17983	12719	4305	601	29	1
25+	1-5 6 7 14 20	3420	19	441	3708	11844	18018	12726	4284	612	27	1
28-	1-5 6 9 10 14	3420	16	461	3652	11928	17948	12754	4284	608	28	1
28+	1-5 6 10 14 17	3420	14	473	3624	11956	17948	12726	4312	596	30	1
30-	1-5 7 8 9 10	3420	18	447	3694	11858	18018	12712	4298	606	28	1
32+	1-5 7 8 11 18	3420	18	448	3687	11879	17983	12747	4277	613	27	1

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 275–308

Tables of Parameters of BIBDs with $r \leq 41$ including Existence, Enumeration, and Resolvability Results

Rudolf Mathon

Department of Computer Science
University of Toronto
Toronto, Ontario, M5S 1A1
CANADA

Alexander Rosa

Department of Mathematical Sciences
McMaster University
Hamilton, Ontario, L8S 4K1
CANADA

1. Introduction

A *balanced incomplete block design* (BIBD) is a pair (V,B) where V is a v-set and B is a collection of b k-subsets of V called *blocks* such that each element of V is contained in exactly r blocks and any 2-subset of V is contained in exactly λ blocks. The numbers v,b,r,k,λ are *parameters* of the BIBD. Trivial necessary conditions for the existence of a BIBD(v,b,r,k,λ) are

(1) $vr = bk$,

(2) $r(k-1) = \lambda(v-1)$

Parameter sets that satisfy (1) and (2) are *admissible*.

A BIBD (V,B) is *resolvable* if there exists a partition R of its set of blocks B into subsets called *parallel classes* each of which in turn partitions the set V; R is called a *resolution*. An additional trivial necessary condition for the existence of a resolvable BIBD is

(3) $k \mid v$.

Two BIBDs (V_1,B_1), (V_2,B_2) are *isomorphic* if there exists a bijection $\alpha: V_1 \to V_2$ such that $B_1\alpha = B_2$. Isomorphism of resolutions of BIBDs is defined similarly.

Given a symmetric BIBD (one with $v = b$, $r = k$), one obtains from it the *residual design* by deleting all elements of one block, and the *derived design* by deleting all elements of the complement of one block. The parameters of the former are $(v-k, v-1, k, k-\lambda, \lambda)$ while those of the latter are $(k, v-1, k-1, \lambda,$

$\lambda-1$). A BIBD is *nontrivial* if $3 \leq k < v$; such designs satisfy Fisher's inequality $b \geq v$. Given a BIBD(v,b,r,k,λ), any BIBD($v,mb,mr,k,m\lambda$) is termed its *m-multiple*. (Repeated blocks are permitted.)

We present here a listing of admissible parameter sets of nontrivial BIBDs with $r \leq 41$ and $k \leq v/2$. The most extensive previous list appears to be that of DiPaola, Wallis and Wallis [D] although another extensive listing of designs classified according to v and k was compiled by Collens (for a brief history of tables and listings of BIBDs, see [D]). However, our present listing differs from that of [D] not only that it extends it up to $r \leq 41$, which more than doubles its size, but also in that it includes information concerning enumeration of BIBDs, and existence and enumeration of resolvable BIBDs. Our sources were, of course, mainly the existing lists. Several recent journal articles and reports provided additional source of information. We adopted the principle of giving only a "minimal set" of references which results, in particular, in an omission of several earlier listings from references. From our point of view, the listings of Hall, Takeuchi, DiPaola-Wallis-Wallis and Kageyama (for resolvable designs) as well as papers by Hanani and Wilson are basic, and are referred to by letters while the remaining references are referred to by numbers. Unlike most of the earlier lists, we include also multiples of known designs; although their existence is trivially implied, information concerning their number and resolvability usually is not.

2. Description of the Tables

The admissible parameter sets of nontrivial BIBDs satisfying $r \leq 41$, $3 \leq k \leq v/2$ and conditions (1), (2) are ordered lexicographically by r, k and λ (in this order). Thus the numbering in our list bears no relation to numbering in any of the earlier listings.

The column *Nd* contains the number $Nd(v,b,r,k,\lambda)$ of pairwise nonisomorphic BIBD(v,b,r,k,λ)'s or the best known lower bound for this number. The column *Nr* contains a dash - if condition (3) is not satisfied. Otherwise it contains the number *Nr* of pairwise nonisomorphic resolutions of BIBD(v,b,r,k,λ)'s or the best known lower bound for this number. Note that *Nr* is not necessarily the number of nonisomorphic resolvable BIBDs as two nonisomorphic resolutions can have isomorphic underlying (resolvable) design. To illustrate the difference, there are 7 nonisomorphic resolutions of BIBD(15,35,7,3,1)'s but only 4 nonisomorphic resolvable BIBD(15,35,7,3,1)'s (see No.14).

The symbol ? indicates that the existence of the corresponding BIBD (resolvable BIBD, respectively) is in doubt.

The meaning of symbols that occur in the column "Comments" is as follows:

m#x : m-multiple of an existing BIBD No.x;

m#x* : m-multiple of No.x which does not exist or whose existence is undecided;

R#x (D#x) : residual (derived) design of No.x which exists;

R#x* (D#x*) : residual (derived) design of No.x which does not exist or whose existence is undecided;

PG (AG) : projective (affine) geometry;

NE1 : BIBD does not exist by Bruck-Ryser theorem;

NE2 : BIBD is a residual of a BIBD that does not exist by Bruck-Ryser theorem, and $\lambda = 1$ or 2;

NE3 : resolvable BIBD does not exist by Bose's condition.

HD : resolvable BIBD$(4t,8t-2,4t-1,2t,2t-1)$ exists as there exists a symmetric (Hadamard) BIBD$(4t-1,4t-1,2t-1,2t-1,t-1)$.

In the column "References", there are no references given for designs that are multiples of known BIBDs. We use often the trivial formula giving $Nd(v,mb,mr,k,m\lambda) \geq n+1$ provided $Nd(v,b,r,k,\lambda) \geq n$, $m \geq 2, n \geq 1$ (and similarly for Nr).

Acknowledgements

We would like to thank J. Bosak, A.E. Brouwer, H. Hanani, N.L. Johnson, D. Jungnickel, W. Kantor, J.Seberry as well as many other colleagues for valuable comments, suggestions and additions to an earlier version of these tables. Research of the authors is supported by the Natural Sciences and Engineering Research Council of Canada under grants numbered A8651 (RM) and A7268 (AR).

Table. (v, b, r, k, λ) designs with r≤41

No	v	b	r	k	λ	Nd	Nr	Comments	References
1	7	7	3	3	1	1	-	PG	[H]
2	9	12	4	3	1	1	1	R#3,AG	[H]
3	13	13	4	4	1	1	-	PG	[H]
4	6	10	5	3	2	1	0	R#7,NE3	[H]
5	16	20	5	4	1	1	1	R#6,AG	[H]
6	21	21	5	5	1	1	-	PG	[H]
7	11	11	5	5	2	1	-		[H]
8	13	26	6	3	1	2	-		[47]
9	7	14	6	3	2	4	-	2#1,D#20	[59]
10	10	15	6	4	2	3	-	R#13	[H,59]
11	25	30	6	5	1	1	1	R#12,AG	[H]
12	31	31	6	6	1	1	-	PG	[H]
13	16	16	6	6	2	3	-		[H,28]
14	15	35	7	3	1	80	7	PG	[H,K,47]
15	8	14	7	4	3	4	1	R#20,AG	[H,K,28,68]
16	15	21	7	5	2	0	0	R#19*,NE2	[H]
17	36	42	7	6	1	0	0	R#18*,NE2,AG	[H]
18	43	43	7	7	1	0	-	NE1,PG	[H]
19	22	22	7	7	2	0	-	NE1	[H]
20	15	15	7	7	3	5	-	PG	[H,59]
21	9	24	8	3	2	36	9	2#2,D#40	[48,58]
22	25	50	8	4	1	≥6	-		[H,15,81]
23	13	26	8	4	2	≥130	-	2#3	[50]
24	9	18	8	4	3	11	-	D#41	[H,28,45]
25	21	28	8	6	2	0	-	R#28*,NE2	[H]
26	49	56	8	7	1	1	1	R#27,AG	[H]
27	57	57	8	8	1	1	-	PG	[H]
28	29	29	8	8	2	0	-	NE1	[H]
29	19	57	9	3	1	≥2395687	-		[H,69]
30	10	30	9	3	2	960	-	D#54	[H,18,27,37]
31	7	21	9	3	3	10	-	3#1	[58]
32	28	63	9	4	1	≥138	≥7		[H,K,12,50]
33.	10	18	9	5	4	21	0	R#41,NE3	[H,28,45]
34	46	69	9	6	1	?	-		
35	16	24	9	6	3	≥26	-	R#40	[H,50]
36	28	36	9	7	2	7	0	R#39,NE3	[H,2]
37	64	72	9	8	1	1	1	R#38,AG	[H]
38	73	73	9	9	1	1	-	PG	[H]
39	37	37	9	9	2	4	-		[H,2]
40	25	25	9	9	3	78	-		[H,25]
41	19	19	9	9	4	6	-		[H,28]
42	21	70	10	3	1	$\geq 2\times 10^6$	≥78		[H,K,47,49,75]
43	6	20	10	3	4	4	1	2#4	[K,30]
44	16	40	10	4	2	≥10	≥10	2#5	[50]
45	41	82	10	5	1	≥1	-		[H]

No	v	b	r	k	λ	Nd	Nr	Comments	References
46	21	42	10	5	2	≥10	-	2#6	[50]
47	11	22	10	5	4	3337	-	2#7,D#63	[37]
48	51	85	10	6	1	?	-		
49	21	30	10	7	3	≥1	0	R#54,NE3	[H,K]
50	36	45	10	8	2	0	-	R#53*,NE2	[H]
51	81	90	10	9	1	≥7	≥7	R#52,AG	[H,22]
52	91	91	10	10	1	≥4	-	PG	[H,22]
53	46	46	10	10	2	0	-	NE1	[H]
54	31	31	10	10	3	≥1	-		[H]
55	12	44	11	3	2	≥574	≥1	D#84	[H,K,50]
56	12	33	11	4	3	≥32	≥1	D#85*	[H,K,20]
57	45	99	11	5	1	≥16	?		[H,50]
58	12	22	11	6	5	601	1	R#63,HD	[H,K,37]
59	45	55	11	9	2	≥11	0	R#62,NE3	[K,23]
60	100	110	11	10	1	?	?	R#61*,AG	
61	111	111	11	11	1	?	-	PG	
62	56	56	11	11	2	≥4	-		[23]
63	23	23	11	11	5	1102	-		[H,36]
64	25	100	12	3	1	$\geq 10^{14}$	-		[H,47]
65	13	52	12	3	2	≥92714	-	2#8,D#96	[50]
66	9	36	12	3	3	≥330	≥10	3#2	[37,50]
67	7	28	12	3	4	35	-	4#1	[30]
68	37	111	12	4	1	≥3	-		[H,19]
69	19	57	12	4	2	≥1	-		[H]
70	13	39	12	4	3	≥198	-	3#3,D#97	[50]
71	10	30	12	4	4	≥15	-	2#10	[80]
72	25	60	12	5	2	≥13	≥13	2#11	[50]
73	61	122	12	6	1	?	-		
74	31	62	12	6	2	≥16	-	2#12	[50]
75	21	42	12	6	3	≥1	-		[H]
76	16	32	12	6	4	≥111	-	2#13	[80]
77	13	26	12	6	5	≥1	-	D#98	[H]
78	22	33	12	8	4	?	-	R#85*	
79	33	44	12	9	3	≥1	-	R#84	[T]
80	55	66	12	10	2	0	-	R#83*,NE2	[H]
81	121	132	12	11	1	≥1	≥1	R#82,AG	[H]
82	133	133	12	12	1	≥1	-	PG	[H]
83	67	67	12	12	2	0	-	NE1	[H]
84	45	45	12	12	3	≥1	-		[T]
85	34	34	12	12	4	0	-	NE1	[H]
86	27	117	13	3	1	$\geq 10^{11}$	≥661	AG	[H,K,38,47]
87	40	130	13	4	1	$\geq 10^{6}$	≥1	PG	[H,K,13]
88	66	143	13	6	1	≥1	?		[24]
89	14	26	13	7	6	≥12	0	R#98,NE3	[K,T,62]
90	27	39	13	9	4	≥9	≥9	R#97,AG	[H,K,13]
91	40	52	13	10	3	?	0	R#96*,NE3	[K]
92	66	78	13	11	2	≥2	0	R#95,NE3	[K,1]
93	144	156	13	12	1	?	?	R#94*,AG	
94	157	157	13	13	1	?	-	PG	
95	79	79	13	13	2	≥2	-		[1]

No	v	b	r	k	λ	Nd	Nr	Comments	References
96	53	53	13	13	3	0	-	NE1	[H]
97	40	40	13	13	4	≥24	-	PG	[H,31]
98	27	27	13	13	6	≥7	-		[H,71]
99	15	70	14	3	2	≥685521	≥21	2#14,D#140	[50]
100	22	77	14	4	2	≥1	-		[H]
101	8	28	14	4	6	2224	4	2#15	[29,50]
102	15	42	14	5	4	≥1	?	2#16*,D141*	[H]
103	36	84	14	6	2	≥1	≥1	2#17*	[H,K]
104	15	35	14	6	5	≥1	-	D#142	[H]
105	85	170	14	7	1	?	-		
106	43	86	14	7	2	≥1	-	2#18*	[H]
107	29	58	14	7	3	≥1	-		[H]
108	22	44	14	7	4	≥1	-	2#19*	[26]
109	15	30	14	7	6	≥6	-	2#20,D#143	
110	78	91	14	12	2	0	-	R#113*,NE2	[H]
111	169	182	14	13	1	≥1	≥1	AG	[H]
112	183	183	14	14	1	≥1	-	PG	[H]
113	92	92	14	14	2	0	-	NE1	[H]
114	31	155	15	3	1	$\geq 2\times 10^{15}$	-		[H,50]
115	16	80	15	3	2	≥4777436	-	D#169	[H,50]
116	11	55	15	3	3	≥29845	-		[H,50]
117	7	35	15	3	5	108	-	5#1	[37]
118	6	30	15	3	6	6	0	3#4	[30,50]
119	16	60	15	4	3	$\geq 6\times 10^{5}$	$\geq 6\times 10^{5}$	3#5,D#170	[50]
120	61	183	15	5	1	≥10	-		[H,19]
121	31	93	15	5	2	≥1	-		[H]
122	21	63	15	5	3	$\geq 10^{9}$	-	3#6	[13]
123	16	48	15	5	4	≥11	-	D#171	[H,14]
124	13	39	15	5	5	≥30	-		[H,72]
125	11	33	15	5	6	≥127	-	3#7	[9,80]
126	76	190	15	6	1	≥1	-		[57]
127	26	65	15	6	3	≥1	-		[H]
128	16	40	15	6	5	≥1	-	D#172	[H]
129	91	195	15	7	1	≥2	?		[H,8]
130	16	30	15	8	7	≥51	≥5	R#143,AG,HD	[H,K,7,37]
131	21	35	15	9	6	$\geq 10^{4}$	-	R#142	[H,15]
132	136	204	15	10	1	?	-		
133	46	69	15	10	3	?	-		
134	28	42	15	10	5	?	-	R#141*	
135	56	70	15	12	3	≥4	-	R#140	[31]
136	91	105	15	13	2	0	0	R#139*,NE2	[H]
137	196	210	15	14	1	0	0	R#138*,NE2,AG	[H]
138	211	211	15	15	1	0	-	NE1,PG	[H]
139	106	106	15	15	2	0	-	NE1	[H]
140	71	71	15	15	3	≥8	-		[31]
141	43	43	15	15	5	0	-	NE1	[H]
142	36	36	15	15	6	≥16448	-		[T,12]
143	31	31	15	15	7	≥1266891	-	PG	[H,60]
144	33	176	16	3	1	$\geq 10^{13}$	≥1		[T,K,50]
145	9	48	16	3	4	≥330	≥9	4#2	[37]

No	v	b	r	k	λ	Nd	Nr	Comments	References
146	49	196	16	4	1	≥224	-		[T,16,19]
147	25	100	16	4	2	≥5	-	2#22	
148	17	68	16	4	3	≥1	-	D#185	[T]
149	13	52	16	4	4	≥198	-	4#3	[50]
150	9	36	16	4	6	≥12	-	2#24	
151	65	208	16	5	1	≥2	≥1		[K,T,16,19]
152	81	216	16	6	1	?	-		
153	21	56	16	6	4	≥1	-	2#25*	[Han]
154	49	112	16	7	2	≥1	≥1	2#26	
155	113	226	16	8	1	?	-		
156	57	114	16	8	2	≥1	-	2#27	
157	29	58	16	8	4	≥1	-	2#28*	[T]
158	17	34	16	8	7	≥1	-	D#186	[T]
159	145	232	16	10	1	?	-		
160	25	40	16	10	6	≥1	-	R#172	[10,73]
161	33	48	16	11	5	≥19	0	R#171,NE3	[K,14]
162	177	236	16	12	1	?	-		
163	45	60	16	12	4	≥1	-	R#170	[4]
164	65	80	16	13	3	?	0	R#169*,NE3	[K]
165	105	120	16	14	2	?	-	R#168*	
166	225	240	16	15	1	?	?	R#167*,AG	
167	241	241	16	16	1	?	-	PG	
168	121	121	16	16	2	?	-		
169	81	81	16	16	3	?	-		
170	61	61	16	16	4	≥1	-		[4]
171	49	49	16	16	5	≥4	-		[14]
172	41	41	16	16	6	≥1	-		[10,73]
173	18	102	17	3	2	$\geq 4\times 10^{14}$	≥1	D#217*	[T,16,33,50]
174	52	221	17	4	1	≥206	≥1		[K,T,19]
175	35	119	17	5	2	≥1	?		[T]
176	18	51	17	6	5	≥3	≥2	D#218*	[T,39,42]
177	35	85	17	7	3	≥1	?		[34,50]
178	120	255	17	8	1	≥1	≥1		[65]
179	18	34	17	9	8	$\geq 10^{3}$	0	R#186,NE3	[K,T,15]
180	52	68	17	13	4	≥1	0	R#185,NE3	[K,D]
181	120	136	17	15	2	0	0	R#184*,NE2	[H]
182	256	272	17	16	1	≥189	≥189	R#183,AG	[H,40,41]
183	273	273	17	17	1	≥13	-	PG	[H,40,41]
184	137	137	17	17	2	0	-	NE1	[H]
185	69	69	17	17	4	≥1	-		[D]
186	35	35	17	17	8	≥1853	-		[T,15]
187	37	222	18	3	1	$\geq 10^{10}$	-		[T,16,44]
188	19	114	18	3	2	$\geq 2\times 10^{9}$	-	2#29,D#231*	[50]
189	13	78	18	3	3	$\geq 3\times 10^{9}$	-	3#8	[50]
190	10	60	18	3	4	≥961	-	2#30	
191	7	42	18	3	6	417	-	6#1	[37]
192	28	126	18	4	2	≥139	≥8	2#32	
193	10	45	18	4	6	≥14819	-	3#10	[50]
194	25	90	18	5	3	$\geq 10^{17}$	$\geq 10^{17}$	3#11	[50]
195	10	36	18	5	8	≥22	≥1	2#33	[K]

No	v	b	r	k	λ	Nd	Nr	Comments	References
196	91	273	18	6	1	≥4	-		[Han,17,54]
197	46	138	18	6	2	≥1	-	2#34*	[Han]
198	31	93	18	6	3	$\geq 10^{22}$	-	3#12	[50]
199	19	57	18	6	5	≥1	-	D#232*	[T]
200	16	48	18	6	6	$\geq 10^{8}$	-	3#13,2#35	[50]
201	28	72	18	7	4	≥392	?	2#36	[80]
202	64	144	18	8	2	≥1	≥1	2#37	
203	145	290	18	9	1	?	-		
204	73	146	18	9	2	≥1	-	2#38	
205	49	98	18	9	3	?	-		
206	37	74	18	9	4	≥852	-	2#39	[80]
207	25	50	18	9	6	≥79	-	2#40	
208	19	38	18	9	8	≥7	-	2#41,D#233	
209	55	99	18	10	3	?	-		
210	100	150	18	12	2	?	-		
211	34	51	18	12	6	?	-	R#218*	
212	85	102	18	15	3	?	-	R#217*	
213	136	153	18	16	2	?	-	R#216*	
214	289	306	18	17	1	≥1	≥1	R#215,AG	[H]
215	307	307	18	18	1	≥1	-	PG	[H]
216	154	154	18	18	2	?	-		
217	103	103	18	18	3	0	-	NE1	[H]
218	52	52	18	18	6	0	-	NE1	[H]
219	39	247	19	3	1	$\geq 10^{44}$	≥88		[K,T,44,70]
220	20	95	19	4	3	≥1	≥1	D#270	[T,3]
221	20	76	19	5	4	≥1	≥1	D#271*	[Han,T,50]
222	96	304	19	6	1	≥1	?		[55]
223	153	323	19	9	1	?	?		
224	20	38	19	10	9	≥32	3	R#233,HD	[6]
225	39	57	19	13	6	?	0	R#232*,NE3	[K]
226	96	114	19	16	3	?	0	R#231*,NE3	[K]
227	153	171	19	17	2	0	0	R#230*,NE2	[H]
228	324	342	19	18	1	?	?	R#229*,AG	
229	343	343	19	19	1	?	-	PG	
230	172	172	19	19	2	0	-	NE1	[H]
231	115	115	19	19	3	?	-		
232	58	58	19	19	6	0	-	NE1	[H]
233	39	39	19	19	9	≥38	-		[6]
234	21	140	20	3	2	$\geq 5\times 10^{14}$	≥79	2#42,D#307	[50]
235	9	60	20	3	5	≥330	≥9	5#2	
236	6	40	20	3	8	13	1	4#4	[37]
237	61	305	20	4	1	≥18132	-		[T,16,19]
238	31	155	20	4	2	≥1	-		[T]
239	21	105	20	4	3	≥1	-	D#308*	[T]
240	16	80	20	4	4	$\geq 6\times 10^{6}$	$\geq 6\times 10^{6}$	4#5	[50]
241	13	65	20	4	5	≥396	-	5#3	[50]
242	11	55	20	4	6	≥1	-		[T]
243	81	324	20	5	1	≥1	-		[T]
244	41	164	20	5	2	≥1	-	2#45	
245	21	84	20	5	4	$\geq 10^{9}$	-	4#6,D#309	[50]

No	v	b	r	k	λ	Nd	Nr	Comments	References
246	17	68	20	5	5	≥1	-		[T,16]
247	11	44	20	5	8	≥3337	-	4#7	
248	51	170	20	6	2	≥1	-	2#48*	[Han]
249	21	70	20	6	5	≥1	-	D#310	[Han]
250	21	60	20	7	6	≥1	?	2#49,D#311*	
251	36	90	20	8	4	?	-	2#50*	
252	81	180	20	9	2	≥8	≥8	2#51	
253	181	362	20	10	1	?	-		
254	91	182	20	10	2	≥5	-	2#52	
255	61	122	20	10	3	?	-		
256	46	92	20	10	4	?	-	2#53*	
257	37	74	20	10	5	≥1	-		[T]
258	31	62	20	10	6	≥1	-	2#54	
259	21	42	20	10	9	≥2	-	D#312	[T,32]
260	111	185	20	12	2	?	-		
261	45	75	20	12	5	?	-		
262	141	188	20	15	2	?	-		
263	57	76	20	15	5	?	-	R#271	
264	36	48	20	15	8	≥1	-		[67]
265	76	95	20	16	4	≥1	-	R#270	[T,16]
266	171	190	20	18	2	?	-	R#269*	
267	361	380	20	19	1	≥1	≥1	R#268,AG	[H]
268	381	381	20	20	1	≥1	-	PG	[H]
269	191	191	20	20	2	?	-		
270	96	96	20	20	4	≥1	-		[T]
271	77	77	20	20	5	0	-	NE1	[H]
272	43	301	21	3	1	$\geq 5\times 10^{64}$	-		[Han,44]
273	22	154	21	3	2	$\geq 3\times 10^{9}$	-	D#336*	[Han,50]
274	15	105	21	3	3	$\geq 10^{15}$	$\geq 10^{11}$	3#14	[50]
275	8	56	21	3	6	≥101	-		[30]
276	7	49	21	3	7	≥9	-	7#1	
277	64	336	21	4	1	≥12048	≥1		[Han,19,35]
278	8	42	21	4	9	≥943	10	3#15	[29,37,50]
279	85	357	21	5	1	≥10	≥1	PG	[H,19]
280	15	63	21	5	6	≥1	?	3#16*	[Han]
281	106	371	21	6	1	≥1	-		[53]
282	36	126	21	6	3	≥1	?	3#17*	[Han]
283	22	77	21	6	5	≥1	-	D#337	[Han]
284	15	56	21	6	7	≥1	-		[Han,66]
285	127	381	21	7	1	?	-		
286	64	192	21	7	2	?	-		
287	43	129	21	7	3	≥1	-	3#18*	[W]
288	22	66	21	7	6	≥1	-	3#19*,D#338*	[Han]
289	19	57	21	7	7	≥1	-		[W]
290	15	45	21	7	9	$\geq 10^{8}$	-	3#20	[50]
291	57	133	21	9	3	?	-		
292	190	399	21	10	1	?	?		
293	22	42	21	11	10	≥2	0	R#312,NE3	[K,T,32]
294	232	406	21	12	1	?	-		
295	274	411	21	14	1	?	-		

No	v	b	r	k	λ	Nd	Nr	Comments	References
296	92	138	21	14	3	?	-		
297	40	60	21	14	7	?	-	R#311*	
298	295	413	21	15	1	?	-		
299	50	70	21	15	6	≥1	-	R#310	[78]
300	64	84	21	16	5	≥157	≥157	R#309,AG	[H,T,79]
301	85	105	21	17	4	?	0	R#308*,NE3	[K]
302	120	140	21	18	3	?	-	R#307*	
303	190	210	21	19	2	?	0	R#306*,NE3	[K]
304	400	420	21	20	1	?	?	R#305*,AG	
305	421	421	21	21	1	?	-	PG	
306	211	211	21	21	2	?	-		
307	141	141	21	21	3	0	-	NE1	[H]
308	106	106	21	21	4	0	-	NE1	[H]
309	85	85	21	21	5	≥213964	-	PG	[H,79]
310	71	71	21	21	6	≥2	-		[78]
311	61	61	21	21	7	0	-	NE1	[H]
312	43	43	21	21	10	≥2	-		[T,32]
313	45	330	22	3	1	$\geq 6\times 10^{76}$	≥84		[Han,44,50]
314	12	88	22	3	4	≥575	≥1	2#55	
315	34	187	22	4	2	≥1	-		[Han]
316	12	66	22	4	6	≥33	≥1	2#56	
317	45	198	22	5	2	≥17	?	2#57	
318	111	407	22	6	1	≥1	-		[55]
319	12	44	22	6	10	≥602	≥400	2#58	[9]
320	133	418	22	7	1	?	?		
321	45	110	22	9	4	≥1353	?	2#59	[80]
322	100	220	22	10	2	?	?	2#60*	
323	221	442	22	11	1	?	-		
324	111	222	22	11	2	?	-	2#61*	
325	56	112	22	11	4	≥2696	-	2#62	[80]
326	45	90	22	11	5	≥80	-		[15]
327	23	46	22	11	10	≥1103	-	2#63,D#351	
328	287	451	22	14	1	?	-		
329	45	66	22	15	7	?	0	R#338*	[22]
330	56	77	22	16	6	≥1	-	R#337	[78]
331	133	154	22	19	3	?	0	R#336*,NE3	[K]
332	210	231	22	20	2	0	-	R#335*,NE2	[H]
333	441	462	22	21	1	0	0	R#334*,NE2,AG	[H]
334	463	463	22	22	1	0	-	NE1,PG	[H]
335	232	232	22	22	2	0	-	NE1	[H]
336	155	155	22	22	3	?	-		
337	78	78	22	22	6	≥1	-		[78]
338	67	67	22	22	7	0	-	NE1	[H]
339	24	184	23	3	2	$\geq 3\times 10^{9}$	≥1	D#404*	[Han,33,50]
340	24	138	23	4	3	≥1	≥1	D#405*	[Han,3]
341	24	92	23	6	5	≥1	?	D#406*	[Han]
342	70	230	23	7	2	?	?		
343	24	69	23	8	7	≥1	?	D#407	[Han]
344	70	161	23	10	3	?	?		
345	231	483	23	11	1	?	?		

No	v	b	r	k	λ	Nd	Nr	Comments	References
346	24	46	23	12	11	≥129	≥129	R#351,HD	[T,36]
347	231	253	23	21	2	0	0	R#350*,NE2	[H]
348	484	506	23	22	1	0	0	R#349*,NE2,AG	[H]
349	507	507	23	23	1	0	-	NE1,PG	[H]
350	254	254	23	23	2	0	-	NE1	[H]
351	47	47	23	23	11	≥1	-		[T]
352	49	392	24	3	1	$\geq 6\times10^{14}$	-		[Han,44]
353	25	200	24	3	2	$\geq 10^{14}$	-	2#64,D#438*	
354	17	136	24	3	3	≥4968	-		[Han,50]
355	13	104	24	3	4	$\geq 10^{8}$	-	4#8	[50]
356	9	72	24	3	6	$\geq 10^{7}$	$\geq 10^{6}$	6#2	[50]
357	7	56	24	3	8	≥35	-	8#1	
358	73	438	24	4	1	$\geq 10^{7}$	-		[11]
359	37	222	24	4	2	≥4	-	2#68	
360	25	150	24	4	3	$\geq 10^{22}$	-	3#22,D#439*	[50]
361	19	114	24	4	4	≥1	-	2#69	
362	13	78	24	4	6	$\geq 10^{8}$	-	6#3	[50]
363	10	60	24	4	8	≥14819	-	4#10	[50]
364	9	54	24	4	9	$\geq 10^{6}$	-	3#24	[50]
365	25	120	24	5	4	$\geq 10^{17}$	$\geq 10^{17}$	4#11,D#440	[50]
366	121	484	24	6	1	≥1	-		[Han,54]
367	61	244	24	6	2	≥1	-	2#73*	[Han]
368	41	164	24	6	3	≥1	-		[Han]
369	31	124	24	6	4	$\geq 10^{22}$	-	4#12	[50]
370	25	100	24	6	5	≥1	-	D#441	[Han]
371	21	84	24	6	6	≥1	-	3#25*,2#75	
372	16	64	24	6	8	$\geq 10^{8}$	-	4#13	[50]
373	13	52	24	6	10	≥1	-	2#77	
374	49	168	24	7	3	$\geq 10^{52}$	$\geq 10^{52}$	3#26	[50]
375	169	507	24	8	1	?	-		
376	85	255	24	8	2	?	-		
377	57	171	24	8	3	$\geq 10^{68}$	-	3#27	[50]
378	43	129	24	8	4	≥1	-		[W]
379	29	87	24	8	6	≥1	-	3#28*	[Han]
380	25	75	24	8	7	≥1	-	D#442*	[W]
381	22	66	24	8	8	≥1	-	2#78*	[Han]
382	33	88	24	9	6	≥1	-	2#79	
383	55	132	24	10	4	?	-	2#80*	
384	25	60	24	10	9	≥3	-	D#443	[W,21]
385	121	264	24	11	2	≥1	≥1	2#81	
386	265	530	24	12	1	?	-		
387	133	266	24	12	2	≥1	-	2#82	
388	89	178	24	12	3	?	-		
389	67	134	24	12	4	?	-	2#83*	
390	45	90	24	12	6	≥1	-	2#84	
391	34	68	24	12	8	?	-	2#85*	
392	25	50	24	12	11	≥10	-	D#444	[T,W,21]
393	105	180	24	14	3	?	-		
394	85	136	24	15	4	?	-		
395	46	69	24	16	8	≥1	-	R#407	[78]

No	v	b	r	k	λ	Nd	Nr	Comments	References
396	69	92	24	18	6	?	-	R#406*	
397	115	138	24	20	4	?	-	R#405*	
398	161	184	24	21	3	?	-	R#404*	
399	49	56	24	21	10	≥1	-		[W]
400	253	276	24	22	2	0	-	R#403*,NE2	[H]
401	529	552	24	23	1	≥1	≥1	R#402,AG	[H]
402	553	553	24	24	1	≥1	-	PG	[H]
403	277	277	24	24	2	0	-	NE1	[H]
404	185	185	24	24	3	0	-	NE1	[H]
405	139	139	24	24	4	?	-		
406	93	93	24	24	6	0	-	NE1	[H]
407	70	70	24	24	8	≥1	-		[78]
408	51	425	25	3	1	$\geq 6\times 10^{53}$	≥9419		[Han,44,70]
409	6	50	25	3	10	19	0	5#4	[37,50]
410	76	475	25	4	1	≥32	≥1		[Han,19,35]
411	16	100	25	4	5	$\geq 10^{6}$	$\geq 10^{6}$	5#5	[50]
412	101	505	25	5	1	≥1	-		[Han]
413	51	255	25	5	2	≥1	-		[Han]
414	26	130	25	5	4	≥1	-	D#471*	[Han]
415	21	105	25	5	5	$\geq 10^{9}$	-	5#6	[50]
416	11	55	25	5	10	≥3337	-	5#7	
417	126	525	25	6	1	≥2	≥1		[Han,22,52]
418	176	550	25	8	1	?	?		
419	226	565	25	10	1	?	-		
420	76	190	25	10	3	?	-		
421	46	115	25	10	5	?	-		
422	26	65	25	10	9	≥1	-	D#472	[Han]
423	276	575	25	12	1	?	?		
424	26	50	25	13	12	≥1	0	R#444,NE3	[K,T]
425	351	585	25	15	1	?	-		
426	51	85	25	15	7	?	-		
427	36	60	25	15	10	≥1	-	R#443	[Han]
428	51	75	25	17	8	?	0	R#442*,NE3	[K]
429	76	100	25	19	6	≥1	0	R#441,NE3	[H,K]
430	476	595	25	20	1	?	-		
431	96	120	25	20	5	≥1	-	R#440	[W]
432	126	150	25	21	4	?	0	R#439*,NE3	[K]
433	176	200	25	22	3	?	0	R#438*,NE3	[K]
434	276	300	25	23	2	?	0	R#437*,NE3	[K]
435	576	600	25	24	1	?	?	R#436*,AG	
436	601	601	25	25	1	?	-	PG	
437	301	301	25	25	2	?	-		
438	201	201	25	25	3	?	-		
439	151	151	25	25	4	0	-	NE1	[H]
440	121	121	25	25	5	≥1	-		[W]
441	101	101	25	25	6	≥1	-		[H]
442	76	76	25	25	8	0	-	NE1	[H]
443	61	61	25	25	10	≥1	-		[W]
444	51	51	25	25	12	≥1	-		[T]
445	27	234	26	3	2	$\geq 10^{11}$	≥662	2#86,D#511*	

No	v	b	r	k	λ	Nd	Nr	Comments	References
446	40	260	26	4	2	$\geq 10^6$	≥1	2#87	
447	14	91	26	4	6	≥1	-		[Han]
448	105	546	26	5	1	≥1	?		[Han]
449	66	286	26	6	2	≥1	?	2#88	
450	27	117	26	6	5	≥1	-	D#512*	[Han]
451	14	52	26	7	12	≥13	≥1	2#89	[H]
452	92	299	26	8	2	?	-		
453	27	78	26	9	8	≥1	≥1	2#90,D#513	
454	235	611	26	10	1	?	-		
455	40	104	26	10	6	?	?	2#91*	
456	66	156	26	11	4	≥494	?	2#92	[80]
457	144	312	26	12	2	?	?	2#93*	
458	313	626	26	13	1	?	-		
459	157	314	26	13	2	?	-	2#94*	
460	105	210	26	13	3	?	-		
461	79	158	26	13	4	≥940	-	2#95	[80]
462	53	106	26	13	6	≥1	-	2#96*	[W]
463	40	80	26	13	8	≥25	-	2#97	
464	27	54	26	13	12	≥8	-	2#98,D#514	
465	40	65	26	16	10	≥1	-	R#472	[73]
466	105	130	26	21	5	?	0	R#471*,NE3	[K]
467	300	325	26	24	2	0	-	R#470*,NE2	[H]
468	625	650	26	25	1	≥33	≥33	R#469,AG	[H,43]
469	651	651	26	26	1	≥17	-	PG	[H,43]
470	326	326	26	26	2	0	-	NE1	[H]
471	131	131	26	26	5	?	-		
472	66	66	26	26	10	≥1	-		[73]
473	55	495	27	3	1	$\geq 6\times 10^{76}$	-		[Han,44]
474	28	252	27	3	2	$\geq 10^{66}$	-	D#564*	[Han,50]
475	19	171	27	3	3	$\geq 10^{17}$	-	3#29	[50]
476	10	90	27	3	6	$\geq 10^{12}$	-	3#30	[50]
477	7	63	27	3	9	≥10	-	9#1	
478	28	189	27	4	3	$\geq 10^{32}$	$\geq 10^{26}$	3#32,D#565*	[50]
479	55	297	27	5	2	≥1	?		[Han]
480	10	54	27	5	12	$\geq 10^{8}$	0	3#33	[50]
481	136	612	27	6	1	≥1	-		[56]
482	46	207	27	6	3	≥1	-	3#34*	[Han]
483	28	126	27	6	5	≥1	-	D#566*	[Han]
484	16	72	27	6	9	≥27	-	3#35	
485	28	108	27	7	6	≥5047	?	3#36,D#567	[80]
486	64	216	27	8	3	$\geq 10^{77}$	$\geq 10^{77}$	3#37	[50]
487	217	651	27	9	1	?	-		
488	109	327	27	9	2	≥1	-		[34,50]
489	73	219	27	9	3	$\geq 10^{90}$	-	3#38	[50]
490	55	165	27	9	4	?	-		
491	37	111	27	9	6	$\geq 10^{37}$	-	3#39	[50]
492	28	84	27	9	8	≥2	-	D#568*	[Han,50]
493	25	75	27	9	9	$\geq 10^{28}$	-	3#40	[50]
494	19	57	27	9	12	$\geq 10^{16}$	-	3#41	[50]
495	55	135	27	11	5	?	?		

No	v	b	r	k	λ	Nd	Nr	Comments	References
496	100	225	27	12	3	?	-		
497	28	63	27	12	11	≥1	-	D#569	[66]
498	325	675	27	13	1	?	?		
499	28	54	27	14	13	≥4	≥4	R#514,HD	[H,T,71]
500	190	342	27	15	2	?	-		
501	55	99	27	15	7	?	-		
502	460	690	27	18	1	?	-		
503	154	231	27	18	3	?	-		
504	52	78	27	18	9	≥1	-	R#513	[W]
505	91	117	27	21	6	?	-	R#512*	
506	208	234	27	24	3	?	-	R#511*	
507	325	351	27	25	2	?	0	R#510*,NE3	[K]
508	676	702	27	26	1	?	?	R#509*,AG	
509	703	703	27	27	1	?	-	PG	
510	352	352	27	27	2	?	-		
511	235	235	27	27	3	0	-	NE1	[H]
512	118	118	27	27	6	0	-	NE1	[H]
513	79	79	27	27	9	≥1	-		[W]
514	55	55	27	27	13	≥1	-		[T,16]
515	57	532	28	3	1	$\geq 10^{90}$	≥1		[Han,35,44]
516	15	140	28	3	4	$\geq 10^{15}$	$\geq 10^{11}$	4#14	[50]
517	9	84	28	3	7	≥330	≥9	7#2	
518	85	595	28	4	1	$\geq 10^{15}$	-		[Han,11]
519	43	301	28	4	2	≥1	-		[Han]
520	29	203	28	4	3	≥1	-	D#586*	[Han]
521	22	154	28	4	4	≥1	-	2#100	
522	15	105	28	4	6	≥1	-		[Han]
523	13	91	28	4	7	$\geq 10^{8}$	-	7#3	[50]
524	8	56	28	4	12	≥2224	31	4#15	[29,37]
525	15	84	28	5	8	≥1	?	4#16*,2#102	
526	141	658	28	6	1	?	-		
527	36	168	28	6	4	≥1	≥1	4#17*,2#103	
528	21	98	28	6	7	≥1	-		[Han]
529	16	70	28	6	10	≥1	-	2#104	
530	169	676	28	7	1	≥1	-		[34,50]
531	85	340	28	7	2	?	-	2#105*	
532	57	228	28	7	3	≥1	-		[Han]
533	43	172	28	7	4	≥1	-	4#18*,2#106	
534	29	116	28	7	6	≥1	-	2#107,D#587*	
535	25	100	28	7	7	≥1	-		[W]
536	22	88	28	7	8	≥1	-	4#19*,2#108	
537	15	60	28	7	12	$\geq 10^{8}$	-	4#20	[50]
538	50	175	28	8	4	?	-		
539	225	700	28	9	1	?	?		
540	85	238	28	10	3	?	-		
541	309	721	28	12	1	?	-		
542	78	182	28	12	4	?	-	2#110*	
543	45	105	28	12	7	?	-		
544	169	364	28	13	2	≥1	≥1	2#111	
545	365	730	28	14	1	?	-		

No	v	b	r	k	λ	Nd	Nr	Comments	References
546	183	366	28	14	2	≥1	-	2#112	
547	92	184	28	14	4	?	-	2#113*	
548	53	106	28	14	7	≥1	-		[W]
549	29	58	28	14	13	≥1	-	D#588	[W]
550	36	63	28	16	12	≥1	-	R#569	[66]
551	477	742	28	18	1	?	-		
552	57	84	28	19	9	?	0	R#568*,NE3	[K]
553	561	748	28	21	1	?	-		
554	141	188	28	21	4	?	-		
555	81	108	28	21	7	≥1	-	R#567	[H]
556	57	76	28	21	10	?	-		
557	99	126	28	22	6	?	-	R#566*	
558	162	189	28	24	4	?	-	R#565*	
559	225	252	28	25	3	?	0	R#564*,NE3	[K]
560	351	378	28	26	2	0	-	R#563*,NE2	[H]
561	729	756	28	27	1	≥7	≥7	R#562,AG	[H,43]
562	757	757	28	28	1	≥3	-	PG	[H,22]
563	379	379	28	28	2	0	-	NE1	[H]
564	253	253	28	28	3	?	-		
565	190	190	28	28	4	0	-	NE1	[H]
566	127	127	28	28	6	?	-		
567	109	109	28	28	7	≥1	-		[H]
568	85	85	28	28	9	?	-		
569	64	64	28	28	12	≥1	-		[66]
570	30	290	29	3	2	$\geq 2\times 10^{51}$	≥1	D#655*	[Han,33,50]
571	88	638	29	4	1	≥2	≥1		[Han,11,35]
572	30	174	29	5	4	≥1	?	D#656	[Han]
573	30	145	29	6	5	≥1	?	D#657*	[Han]
574	175	725	29	7	1	?	?		
575	117	377	29	9	2	≥1	?		[34]
576	30	87	29	10	9	?	?	D#658*	
577	117	261	29	13	3	?	?		
578	378	783	29	14	1	?	?		
579	30	58	29	15	14	≥1	0	R#588,NE3	[K,W]
580	88	116	29	22	7	?	0	R#587*,NE3	[K]
581	175	203	29	25	4	?	0	R#586*,NE3	[K]
582	378	406	29	27	2	?	0	R#585*,NE3	[K]
583	784	812	29	28	1	?	?	R#584*,AG	
584	813	813	29	29	1	?	-	PG	
585	407	407	29	29	2	?	-		
586	204	204	29	29	4	?	-		
587	117	117	29	29	7	0	-	NE1	[H]
588	59	59	29	29	14	≥1	-		[W]
589	61	610	30	3	1	$\geq 2\times 10^{24}$	-		[Han,44]
590	31	310	30	3	2	$\geq 2\times 10^{16}$	-	2#114,D#677*	
591	21	210	30	3	3	$\geq 10^{24}$	$\geq 10^{21}$	3#42	[50]
592	16	160	30	3	4	$\geq 4\times 10^{6}$	-	2#115	
593	13	130	30	3	5	$\geq 10^{8}$	-	5#8	[50]
594	11	110	30	3	6	$\geq 10^{4}$	-	2#116	
595	7	70	30	3	10	≥108	-	10#1	

No	v	b	r	k	λ	Nd	Nr	Comments	References
596	6	60	30	3	12	34	1	6#4,3#43	[37,50]
597	46	345	30	4	2	≥1	-		[Han]
598	16	120	30	4	6	$\geq 10^{15}$	$\geq 10^{15}$	6#5	[50]
599	10	75	30	4	10	≥29638	-	5#10	[50]
600	121	726	30	5	1	≥1	-		[Han]
601	61	366	30	5	2	≥11	-	2#120	
602	41	246	30	5	3	≥1	-	3#45	
603	31	186	30	5	4	≥1	-	2#121,D#678*	
604	25	150	30	5	5	$\geq 10^{17}$	$\geq 10^{17}$	5#11	[50]
605	21	126	30	5	6	$\geq 10^{24}$	-	6#6	[50]
606	16	96	30	5	8	≥12	-	2#123	
607	13	78	30	5	10	≥31	-	2#124	
608	11	66	30	5	12	$\geq 10^{6}$	-	6#7	[50]
609	151	755	30	6	1	≥1	-		[Han]
610	76	380	30	6	2	≥1	-	2#126	
611	51	255	30	6	3	≥1	-	3#48*	[Han]
612	31	155	30	6	5	$\geq 10^{23}$	-	5#12,D#679	[50]
613	26	130	30	6	6	≥1	-	2#127	
614	16	80	30	6	10	$\geq 10^{8}$	-	5#13	[50]
615	91	390	30	7	2	≥3	?	2#129	
616	21	90	30	7	9	≥1	?	3#49	
617	36	135	30	8	6	?	-	3#50*	
618	16	60	30	8	14	≥52	≥6	2#130	
619	81	270	30	9	3	$\geq 10^{108}$	$\geq 10^{108}$	3#51	[50]
620	21	70	30	9	12	$\geq 10^{4}$	-	2#131	
621	271	813	30	10	1	?	-		
622	136	408	30	10	2	?	-	2#132*	
623	91	273	30	10	3	$\geq 10^{125}$	-	3#52	[50]
624	55	165	30	10	5	≥1	-		[34]
625	46	138	30	10	6	?	-	2#133*,3#53*	
626	31	93	30	10	9	≥1	-	3#54,D#680*	
627	28	84	30	10	10	≥2	-	2#134*	[Han,50]
628	166	415	30	12	2	?	-		
629	56	140	30	12	6	≥5	-	2#135	
630	34	85	30	12	10	?	-		
631	91	210	30	13	4	?	?	2#136*	
632.	196	420	30	14	2	?	?	2#137*	
633	421	842	30	15	1	?	-		
634	211	422	30	15	2	?	-	2#138*	
635	141	282	30	15	3	?	-		
636	106	212	30	15	4	?	-	2#139*	
637	85	170	30	15	5	?	-		
638	71	142	30	15	6	≥9	-	2#140	
639	61	122	30	15	7	≥1	-		[W]
640	43	86	30	15	10	≥1	-	2#141*	[Han]
641	36	72	30	15	12	$\geq 10^{4}$	-	2#142	
642	31	62	30	15	14	$\geq 10^{6}$	-	2#143,D#681	
643	171	285	30	18	3	?	-		
644	286	429	30	20	2	?	-		
645	96	144	30	20	6	?	-		

No	v	b	r	k	λ	Nd	Nr	Comments	References
646	58	87	30	20	10	?	-	R#:'658*	
647	301	430	30	21	2	?	-		
648	116	145	30	24	6	?	-	R#657*	
649	145	174	30	25	5	≥1	-	R#656	[W]
650	261	290	30	27	3	?	-	R#655*	
651	406	435	30	28	2	0	-	R#654*,NE2	[H]
652	841	870	30	29	1	≥1	≥1	R#653,AG	[H]
653	871	871	30	30	1	≥1	-	PG	[H]
654	436	436	30	30	2	0	-	NE1	[H]
655	291	291	30	30	3	?	-		
656	175	175	30	30	5	≥1	-		[W]
657	146	146	30	30	6	0	-	NE1	[H]
658	88	88	30	30	10	0	-	NE1	[H]
659	63	651	31	3	1	$\geq 10^{42}$	$\geq 10^{4}$	PG	[H,44,50]
660	32	248	31	4	3	≥1	≥1	D#729*	[Han,3]
661	125	775	31	5	1	≥1	≥1	AG	[H]
662	156	806	31	6	1	≥1	≥1	PG	[H]
663	63	279	31	7	3	≥1	?		[Han]
664	32	124	31	8	7	≥1	≥1	D#730*	[Han,61]
665	63	217	31	9	4	?	?		
666	280	868	31	10	1	?	?		
667	435	899	31	15	1	?	?		
668	32	62	31	16	15	≥20	≥20	R#681,AG,HD	[H,7,32]
669	63	93	31	21	10	?	0	R#680*	[22]
670	125	155	31	25	6	$\geq 10^{12}$	$\geq 10^{12}$	R#679,AG	[H,5,32,79]
671	156	186	31	26	5	?	0	R#678*,NE3	[K]
672	280	310	31	28	3	?	0	R#677*,NE3	[K]
673	435	465	31	29	2	0	0	R#676*,NE2	[H]
674	900	930	31	30	1	0	0	R#675*,NE2,AG	[H]
675	931	931	31	31	1	0	-	NE1,PG	[H]
676	466	466	31	31	2	0	-	NE1	[H]
677	311	311	31	31	3	?	-		
678	187	187	31	31	5	0	-	NE1	[H]
679	156	156	31	31	6	$\geq 10^{17}$	-	PG	[H,5,32,79]
680	94	94	31	31	10	0	-	NE1	[H]
681	63	63	31	31	15	$\geq 10^{17}$	-	PG	[H,7,32,79]
682	33	352	32	3	2	$\geq 10^{18}$	≥1	2#144,D#775*	
683	9	96	32	3	8	$\geq 10^{7}$	$\geq 10^{5}$	8#2	[50]
684	97	776	32	4	1	≥2	-		[11]
685	49	392	32	4	2	≥225	-	2#146	
686	33	264	32	4	3	≥1	-	D#776*	[Han]
687	25	200	32	4	4	$\geq 10^{22}$	-	4#22	[50]
688	17	136	32	4	6	≥1	-	2#148	
689	13	104	32	4	8	$\geq 10^{8}$	-	8#3	[50]
690	9	72	32	4	12	$\geq 10^{6}$	-	4#24	[50]
691	65	416	32	5	2	≥3	≥1	2#151	
692	81	432	32	6	2	≥1	-	2#152*	[Han]
693	33	176	32	6	5	≥1	-	D#777*	[Han]
694	21	112	32	6	8	≥1	-	4#25*,2#153	
695	49	224	32	7	4	$\geq 10^{52}$	$\geq 10^{52}$	4#26	[50]

No	v	b	r	k	λ	Nd	Nr	Comments	References
696	225	900	32	8	1	?	-		
697	113	452	32	8	2	?	-	2#155*	
698	57	228	32	8	4	$\geq 10^{68}$	-	4#27	[50]
699	33	132	32	8	7	≥1	-	D#778	[Han]
700	29	116	32	8	8	≥1	-	4#28*,2#157	
701	17	68	32	8	14	≥1	-	2#158	
702	145	464	32	10	2	?	-	2#159*	
703	25	80	32	10	12	≥1	-	2#160	
704	33	96	32	11	10	≥20	?	2#161,D#779*	
705	177	472	32	12	2	?	-	2#162*	
706	45	120	32	12	8	≥1	-	2#163	
707	33	88	32	12	11	≥1	-	D#780*	[Han]
708	65	160	32	13	6	?	?	2#164*	
709	105	240	32	14	4	?	-	2#165*	
710	225	480	32	15	2	?	?	2#166*	
711	481	962	32	16	1	?	-		
712	241	482	32	16	2	?	-	2#167*	
713	161	322	32	16	3	?	-		
714	121	242	32	16	4	?	-	2#168*	
715	97	194	32	16	5	?	-		
716	81	162	32	16	6	?	-	2#169*	
717	61	122	32	16	8	≥1	-	2#170	
718	49	98	32	16	10	≥5	-	2#171	
719	41	82	32	16	12	≥1	-	2#172	
720	33	66	32	16	15	≥1	-	D#781	[74]
721	305	488	32	20	2	?	-		
722	369	492	32	24	2	?	-		
723	93	124	32	24	8	?	-	R#730*	
724	217	248	32	28	4	?	-	R#729*	
725	465	496	32	30	2	0	-	R#728*,NE2	[H]
726	961	992	32	31	1	≥1	≥1	R#727,AG	[H]
727	993	993	32	32	1	≥1	-	PG	[H]
728	497	497	32	32	2	0	-	NE1	[H]
729	249	249	32	32	4	?	-		
730	125	125	32	32	8	0	-	NE1	[H]
731	67	737	33	3	1	$\geq 10^{35}$	-		[Han,44]
732	34	374	33	3	2	$\geq 10^{41}$	-	D#811*	[Han,50]
733	23	253	33	3	3	$\geq 3\times 10^{7}$	-		[Han,50]
734	12	132	33	3	6	$\geq 10^{8}$	≥1	3#55	[50]
735	7	77	33	3	11	≥107	-	11#1	
736	100	825	33	4	1	≥2	≥1		[Han,11,35]
737	12	99	33	4	9	≥33	≥1	3#56	
738	45	297	33	5	3	$\geq 10^{64}$	?	3#57	[50]
739	166	913	33	6	1	?	-		
740	56	308	33	6	3	≥1	-		[Han]
741	34	187	33	6	5	≥1	-	D#812*	[Han]
742	16	88	33	6	11	≥1	-		[Han]
743	12	66	33	6	15	≥602	≥12	3#58	
744	232	957	33	8	1	≥1	?		[77]
745	45	165	33	9	6	≥35805	?	3#59	[80]

No	v	b	r	k	λ	Nd	Nr	Comments	References
746	100	330	33	10	3	?	?	3#60*	
747	331	993	33	11	1	?	-		
748	166	498	33	11	2	?	-		
749	111	333	33	11	3	?	-	3#61*	
750	67	201	33	11	5	≥1	-		[W]
751	56	168	33	11	6	$\geq 10^{71}$	-	3#62	[50]
752	34	102	33	11	10	≥1	-	D#813*	[Han]
753	31	93	33	11	11	≥1	-		[Han]
754	23	69	33	11	15	≥1103	-	3#63	
755	364	1001	33	12	1	?	-		
756	155	341	33	15	3	?	-		
757	496	1023	33	16	1	≥1	≥1		[65]
758	34	66	33	17	16	≥1	0	R#781,NE3	[K,74]
759	133	209	33	21	5	?	-		
760	56	88	33	21	12	?	-	R#780*	
761	694	1041	33	22	1	?	-		
762	232	348	33	22	3	?	-		
763	100	150	33	22	7	?	-		
764	78	117	33	22	9	?	-		
765	64	96	33	22	11	≥1	-	R#779*	[W]
766	760	1045	33	24	1	?	-		
767	100	132	33	25	8	≥1	0	R#778,NE3	[K,32]
768	144	176	33	27	6	?	-	R#777*	
769	232	264	33	29	4	?	0	R#776*,NE3	[K]
770	320	352	33	30	3	?	-	R#775*	
771	496	528	33	31	2	?	0	R#774*,NE3	[K]
772	1024	1056	33	32	1	≥11	≥11	R#773,AG	[H,22]
773	1057	1057	33	33	1	≥6	-	PG	[H,22]
774	529	529	33	33	2	?	-		
775	353	353	33	33	3	0	-	NE1	[H]
776	265	265	33	33	4	?	-		
777	177	177	33	33	6	?	-		
778	133	133	33	33	8	≥1	-		[32]
779	97	97	33	33	11	?	-		
780	89	89	33	33	12	0	-	NE1	[H]
781	67	67	33	33	16	≥1	-		[74]
782	69	782	34	3	1	$\geq 4\times 10^{41}$	≥1		[Han,35,44]
783	18	204	34	3	4	$\geq 4\times 10^{14}$	≥1	2#173	
784	52	442	34	4	2	≥207	≥1	2#174	
785	18	153	34	4	6	≥1	-		[Han]
786	35	238	34	5	4	≥1	?	2#175,D#869*	
787	171	969	34	6	1	?	-		
788	18	102	34	6	10	≥4	≥3	2#176	
789	35	170	34	7	6	≥2	?	2#177,D#870*	
790	120	510	34	8	2	≥1	≥1	2#178	
791	18	68	34	9	16	$\geq 10^{3}$	≥1	2#179	[H]
792	35	119	34	10	9	≥1	-	D#871*	[Han]
793	341	1054	34	11	1	?	?		
794	52	136	34	13	8	≥1	?	2#180	
795	35	85	34	14	13	?	-	D#872*	

No	v	b	r	k	λ	Nd	Nr	Comments	References
796	120	272	34	15	4	?	?	2#181*	
797	256	544	34	16	2	≥190	≥190	2#182	
798	545	1090	34	17	1	?	-		
799	273	546	34	17	2	≥14	-	2#183	
800	137	274	34	17	4	?	-	2#184*	
801	69	138	34	17	8	≥1	-	2#185	
802	35	70	34	17	16	≥1854	-	2#186,D#873	
803	715	1105	34	22	1	?	-		
804	69	102	34	23	11	?	0	R#813*,NE3	[K]
805	154	187	34	28	6	?	-	R#812*	
806	341	374	34	31	3	?	0	R#811*,NE3	[K]
807	528	561	34	32	2	0	-	R#810*,NE2	[H]
808	1089	1122	34	33	1	0	0	R#809*,NE2,AG	[H]
809	1123	1123	34	34	1	0	-	NE1	[H]
810	562	562	34	34	2	0	-	NE1	[H]
811	375	375	34	34	3	?	-		
812	188	188	34	34	6	0	-	NE1	[H]
813	103	103	34	34	11	?	-		
814	36	420	35	3	2	$\geq 2\times 10^{50}$	≥1	D#961*	[Han,33,50]
815	15	175	35	3	5	$\geq 10^{16}$	$\geq 10^{11}$	5#14	[50]
816	6	70	35	3	14	48	0	7#4	[37,50]
817	36	315	35	4	3	≥1	≥1	D#962*	[Han,3]
818	16	140	35	4	7	$\geq 10^{15}$	$\geq 10^{16}$	7#5	[50]
819	8	70	35	4	15	≥2224	≥5	5#15	
820	141	987	35	5	1	≥1	-		[Han]
821	71	497	35	5	2	≥1	-		[Han]
822	36	252	35	5	4	≥1	-	D#963*	[Han]
823	29	203	35	5	5	≥1	-		[Han]
824	21	147	35	5	7	$\geq 10^{24}$	-	7#6	[50]
825	15	105	35	5	10	≥1	?	5#16*	[Han]
826	11	77	35	5	14	$\geq 10^{6}$	-	7#7	[50]
827	36	210	35	6	5	≥1	≥1	5#17*,D#964*	[Han,3]
828	211	1055	35	7	1	?	-		
829	106	530	35	7	2	?	-	-	
830	71	355	35	7	3	≥1	-		[W]
831	43	215	35	7	5	≥1	-	5#18*	[Han]
832	36	180	35	7	6	≥1	-	D#965*	[Han]
833	31	155	35	7	7	≥1	-	PG	[H]
834	22	110	35	7	10	≥1	-	5#19*	[Han]
835	16	80	35	7	14	≥1	-		[Han]
836	15	75	35	7	15	$\geq 10^{9}$	-	5#20	[50]
837	36	140	35	9	8	≥1	?	D#966*	[Han]
838	316	1106	35	10	1	?	-		
839	106	371	35	10	3	?	-		
840	64	224	35	10	5	?	-		
841	46	161	35	10	7	?	-		
842	36	126	35	10	9	≥1	-	D#967*	[Han]
843	22	77	35	10	15	≥1	-		[Han]
844	176	560	35	11	2	?	?		
845	36	105	35	12	11	≥1	≥1	D#968*	[Han,51]

No	v	b	r	k	λ	Nd	Nr	Comments	References
846	456	1140	35	14	1	?	-		
847	92	230	35	14	5	?	-		
848	66	165	35	14	7	?	-		
849	36	90	35	14	13	≥1	-	D#969*	[Han]
850	246	574	35	15	2	?	-		
851	99	231	35	15	5	?	-		
852	36	84	35	15	14	≥1	-	D#970*	[Han]
853	176	385	35	16	3	?	?		
854	561	1155	35	17	1	?	?		
855	36	70	35	18	17	≥91	≥91	R#873,HD	[15,74]
856	96	168	35	20	7	?	-		
857	351	585	35	21	2	?	-		
858	141	235	35	21	5	?	-		
859	51	85	35	21	14	?	-	R#872*	
860	85	119	35	25	10	?	-	R#871*	
861	316	395	35	28	3	?	-		
862	136	170	35	28	7	?	-	R#870*	
863	64	80	35	28	15	?	-		
864	204	238	35	30	5	?	-	R#869*	
865	561	595	35	33	2	0	0	R#868*,NE2	[H]
866	1156	1190	35	34	1	?	?	R#867*,AG	
867	1191	1191	35	35	1	?	-	PG	
868	596	596	35	35	2	0	-	NE1	[H]
869	239	239	35	35	5	?	-		
870	171	171	35	35	7	?	-		
871	120	120	35	35	10	?	-		
872	86	86	35	35	14	0	-	NE1	[H]
873	71	71	35	35	17	≥1	-		[74]
874	73	876	36	3	1	$\geq 10^{34}$	-		[Han,44]
875	37	444	36	3	2	$\geq 10^{10}$	-	2#187,D#991*	
876	25	300	36	3	3	$\geq 10^{26}$	-	3#64	[50]
877	19	228	36	3	4	$\geq 10^{17}$	-	4#29	[50]
878	13	156	36	3	6	$\geq 10^{17}$	-	6#8	[50]
879	10	120	36	3	8	$\geq 10^{12}$	-	4#30	[50]
880	9	108	36	3	9	≥330	≥10	9#2	
881	7	84	36	3	12	≥417	-	12#1	
882	109	981	36	4	1	≥2	-		[11]
883	55	495	36	4	2	≥1	-		[Han]
884	37	333	36	4	3	$\geq 10^{40}$	-	3#68,D#992*	[50]
885	28	252	36	4	4	$\geq 10^{32}$	$\geq 10^{26}$	4#32	[50]
886	19	171	36	4	6	≥1	-	3#69	
887	13	117	36	4	9	$\geq 10^{8}$	-	9#3	[50]
888	10	90	36	4	12	$\geq 10^{9}$	-	6#10	[50]
889	145	1044	36	5	1	≥1	?		[Han]
890	25	180	36	5	6	$\geq 10^{38}$	$\geq 10^{38}$	6#11	[50]
891	10	72	36	5	16	$\geq 10^{8}$	≥1	4#33,2#195	[50]
892	181	1086	36	6	1	≥1	-		[Han]
893	91	546	36	6	2	≥5	-	2#196	
894	61	366	36	6	3	≥1	-	3#73*	[Han]
895	46	276	36	6	4	≥1	-	4#34*,2#197	

No	v	b	r	k	λ	Nd	Nr	Comments	References
896	37	222	36	6	5	≥1	-	D#993	[Han]
897	31	186	36	6	6	$\geq 10^{61}$	-	6#12	[50]
898	21	126	36	6	9	≥1	-	3#75	
899	19	114	36	6	10	≥1	-	2#199	
900	16	96	36	6	12	$\geq 10^{19}$	-	6#13	[50]
901	13	78	36	6	15	$\geq 10^{6}$	-	3#77	[50]
902	217	1116	36	7	1	≥1	?		[46]
903	28	144	36	7	8	≥5432	?	4#36	[80]
904	64	288	36	8	4	$\geq 10^{77}$	$\geq 10^{77}$	4#37	[50]
905	22	99	36	8	12	≥1	-	3#78*	[Han]
906	289	1156	36	9	1	?	-		
907	145	580	36	9	2	?	-	2#203*	
908	97	388	36	9	3	?	-		
909	73	292	36	9	4	$\geq 10^{90}$	-	4#38	[50]
910	49	196	36	9	6	≥1	-	2#205*	[34,50]
911	37	148	36	9	8	$\geq 10^{37}$	-	4#39,D#994*	[50]
912	33	132	36	9	9	≥1	-	3#79	
913	25	100	36	9	12	$\geq 10^{28}$	-	4#40	
914	19	76	36	9	16	$\geq 10^{16}$	-	4#41	[50]
915	325	1170	36	10	1	?	-		
916	55	198	36	10	6	?	-	3#80*,2#209*	
917	121	396	36	11	3	$\geq 10^{188}$	$\geq 10^{188}$	3#81	[50]
918	397	1191	36	12	1	?	-		
919	199	597	36	12	2	?	-		
920	133	399	36	12	3	$\geq 10^{208}$	-	3#82	[50]
921	100	300	36	12	4	?	-	2#210*	
922	67	201	36	12	6	≥1	-	3#83*	[W]
923	45	135	36	12	9	≥1	-	3#84	
924	37	111	36	12	11	≥1	-	D#995*	[Han]
925	34	102	36	12	12	?	-	3#85*,2#211*	
926	469	1206	36	14	1	?	-		
927	505	1212	36	15	1	?	-		
928	85	204	36	15	6	?	-	2#212*	
929	136	306	36	16	4	?	-	2#213*	
930	289	612	36	17	2	≥1	≥1	2#214	
931	613	1226	36	18	1	?	-		
932	307	614	36	18	2	≥1	-	2#215	
933	205	410	36	18	3	?	-		
934	154	308	36	18	4	?	-	2#216*	
935	103	206	36	18	6	?	-	2#217*	
936	69	138	36	18	9	?	-		
937	52	104	36	18	12	?	-	2#218*	
938	37	74	36	18	17	≥1	-	D#996	[Han]
939	685	1233	36	20	1	?	-		
940	115	207	36	20	6	?	-		
941	721	1236	36	21	1	?	-		
942	91	156	36	21	8	?	-		
943	49	84	36	21	15	?	-	R#970*	
944	253	414	36	22	3	?	-		
945	55	90	36	22	14	?	-	R#969*	

No	v	b	r	k	λ	Nd	Nr	Comments	References
946	208	312	36	24	4	?	-		
947	70	105	36	24	12	?	-	R#968*	
948	91	126	36	26	10	?	-	R#967*	
949	105	140	36	27	9	?	-	R#966*	
950	973	1251	36	28	1	?	-		
951	145	180	36	29	7	?	0	R#965*,NE3	[K]
952	1045	1254	36	30	1	?	-		
953	175	210	36	30	6	?	-	R#964*	
954	217	252	36	31	5	?	0	R#963*,NE3	[K]
955	280	315	36	32	4	?	-	R#962*	
956	385	420	36	33	3	?	-	R#961*	
957	595	630	36	34	2	?	-	R#960*	
958	1225	1260	36	35	1	?	?	R#959*,AG	
959	1261	1261	36	36	1	?	-	PG	
960	631	631	36	36	2	?	-		
961	421	421	36	36	3	?	-		
962	316	316	36	36	4	0	-	NE1	[H]
963	253	253	36	36	5	?	-		
964	211	211	36	36	6	0	-	NE1	[H]
965	181	181	36	36	7	?	-		
966	141	141	36	36	9	?	-		
967	127	127	36	36	10	?	-		
968	106	106	36	36	12	0	-	NE1	[H]
969	91	91	36	36	14	0	-	NE1	[H]
970	85	85	36	36	15	?	-		
971	75	925	37	3	1	$\geq 10^{198}$	≥1		[Han,32,37]
972	112	1036	37	4	1	≥2	≥1		[Han,10,32]
973	75	555	37	5	2	≥1	?		[Han]
974	186	1147	37	6	1	≥1	?		[Han]
975	112	592	37	7	2	≥1	?		[Han]
976	297	1221	37	9	1	?	?		
977	408	1258	37	12	1	?	?		
978	75	185	37	15	7	≥1	≥1		[61]
979	112	259	37	16	5	?	?		
980	630	1295	37	18	1	?	?		
981	38	74	37	19	18	≥1	0	R#996,NE3	[K,74]
982	75	111	37	25	12	?	0	R#995*,NE3	[K]
983	112	148	37	28	9	?	?	R#994*	
984	186	222	37	31	6	≥1	0	R#993,NE3	[K,63]
985	297	333	37	33	4	?	0	R#992*,NE3	[K]
986	408	444	37	34	3	?	0	R#991*,NE3	[K]
987	630	666	37	35	2	0	0	R#990*,NE2	[H]
988	1296	1332	37	36	1	?	?	R#989*,AG	
989	1333	1333	37	37	1	?	-	PG	
990	667	667	37	37	2	0	-	NE1	[H]
991	445	445	37	37	3	0	-	NE1	[H]
992	334	334	37	37	4	0	-	NE1	[H]
993	223	223	37	37	6	≥1	-		[63]
994	149	149	37	37	9	?	-		
995	112	112	37	37	12	?	-		

No	v	b	r	k	λ	Nd	Nr	Comments	References
996	75	75	37	37	18	≥1	-		[64]
997	39	494	38	3	2	$\geq 10^{44}$	≥89	2#219,D#1071*	
998	58	551	38	4	2	≥1	-		[Han]
999	20	190	38	4	6	≥1	≥1	2#220	
1000	20	152	38	5	8	≥1	≥1	2#221	
1001	96	608	38	6	2	≥1	?	2#222	
1002	39	247	38	6	5	≥1	-	D#1072*	[Han]
1003	77	418	38	7	3	≥1	?		[Han]
1004	20	95	38	8	14	≥1	-		[Han]
1005	153	646	38	9	2	≥1	?	2#223*	[34,50]
1006	115	437	38	10	3	?	-		
1007	20	76	38	10	18	≥33	≥4	2#224	
1008	77	266	38	11	5	?	?		
1009	210	665	38	12	2	?	-		
1010	39	114	38	13	12	≥1	?	2#225*,D#1073*	[Han]
1011	96	228	38	16	6	?	?	2#226*	
1012	153	342	38	17	4	?	?	2#227*	
1013	324	684	38	18	2	?	?	2#228*	
1014	685	1370	38	19	1	?	-		
1015	343	686	38	19	2	?	-	2#229*	
1016	229	458	38	19	3	?	-		
1017	172	344	38	19	4	?	-	2#230*	
1018	115	230	38	19	6	?	-	2#231*	
1019	77	154	38	19	9	?	-		
1020	58	116	38	19	12	?	-	2#232*	
1021	39	78	38	19	18	≥39	-	2#233,D#1074	
1022	666	703	38	36	2	?	-	R#1025*	
1023	1369	1406	38	37	1	≥1	≥1	R#1024,AG	[H]
1024	1407	1407	38	38	1	≥1	-	PG	[H]
1025	704	704	38	38	2	?	-		
1026	79	1027	39	3	1	$\geq 10^{56}$	-		[Han,44]
1027	40	520	39	3	2	$\geq 6\times 10^{24}$	-	D#1163*	[Han,50]
1028	27	351	39	3	3	$\geq 10^{28}$	$\geq 10^{30}$	3#86	[50]
1029	14	182	39	3	6	$\geq 2\times 10^{34}$	-		[Han,50]
1030	7	91	39	3	13	≥417	-	13#1	
1031	40	390	39	4	3	$\geq 10^{33}$	$\geq 10^{33}$	3#87,D#1164*	[50]
1032	40	312	39	5	4	≥1	?	D#1165*	[Han]
1033	196	1274	39	6	1	?	?		
1034	66	429	39	6	3	≥1	≥1	3#88	[64]
1035	40	260	39	6	5	≥1	-	D#1166*	[Han]
1036	16	104	39	6	13	≥1	-		[Han]
1037	14	91	39	6	15	≥1	-		[Han]
1038	14	78	39	7	18	≥13	0	3#89	[50]
1039	40	195	39	8	7	≥1	?	D#1167*	[Han]
1040	105	455	39	9	3	?	-		
1041	27	117	39	9	12	$\geq 10^{17}$	$\geq 10^{17}$	3#90	[50]
1042	40	156	39	10	9	≥1	?	3#91*,D#1168*	[Han]
1043	66	234	39	11	6	≥21584	?	3#92	[80]
1044	144	468	39	12	3	?	?	3#93*	
1045	40	130	39	12	11	≥1	-	D#1169*	[Han]

No	v	b	r	k	λ	Nd	Nr	Comments	References
1046	469	1407	39	13	1	?	-		
1047	235	705	39	13	2	?	-		
1048	157	471	39	13	3	?	-	3#94*	
1049	118	354	39	13	4	?	-		
1050	79	237	39	13	6	$\geq 10^{125}$	-	3#95	[50]
1051	53	159	39	13	9	?	-	3#96*	
1052	40	120	39	13	12	$\geq 10^{33}$	-	3#97,D#1170	[50]
1053	37	111	39	13	13	≥1	-		[Han]
1054	27	81	39	13	18	≥8	-	3#98	
1055	40	104	39	15	14	≥1	-	D#1171*	[Han]
1056	222	481	39	18	3	?	-		
1057	703	1443	39	19	1	?	?		
1058	40	78	39	20	19	≥1	≥1	R#1074,HD	[74]
1059	196	364	39	21	4	?	-		
1060	976	1464	39	26	1	?	-		
1061	326	489	39	26	3	?	-		
1062	196	294	39	26	5	?	-		
1063	76	114	39	26	13	?	-	R#1073*	
1064	66	99	39	26	15	?	-		
1065	209	247	39	33	6	?	-	R#1072*	
1066	456	494	39	36	3	?	-	R#1071*	
1067	703	741	39	37	2	0	0	R#1070*,NE2	[H]
1068	1444	1482	39	38	1	0	0	R#1069*,NE2,AG	[H]
1069	1483	1483	39	39	1	0	-	NE1,PG	[H]
1070	742	742	39	39	2	0	-	NE1	[H]
1071	495	495	39	39	3	?	-		
1072	248	248	39	39	6	0	-	NE1	[H]
1073	115	115	39	39	13	0	-	NE1	[H]
1074	79	79	39	39	19	≥1	-		[64]
1075	81	1080	40	3	1	$\geq 10^{48}$	$\geq 10^{7}$	AG	[H,44,50]
1076	21	280	40	3	4	$\geq 10^{24}$	$\geq 10^{21}$	4#42	[50]
1077	9	120	40	3	10	$\geq 10^{8}$	$\geq 10^{5}$	10#2	[50]
1078	6	80	40	3	16	76	1	8#4,4#43	[50]
1079	121	1210	40	4	1	$\geq 10^{6}$	-		[11,13]
1080	61	610	40	4	2	$\geq 10^{4}$	-	2#237	
1081	41	410	40	4	3	≥1	-	D#1192*	
1082	31	310	40	4	4	≥1	-	2#238	
1083	25	250	40	4	5	$\geq 10^{23}$	-	5#22	[50]
1084	21	210	40	4	6	≥1	-	2#239	
1085	16	160	40	4	8	$\geq 10^{15}$	≥1	8#5	[50]
1086	13	130	40	4	10	$\geq 10^{14}$	-	10#3	[50]
1087	11	110	40	4	12	≥1	-	2#242	
1088	9	90	40	4	15	$\geq 10^{6}$	-	5#24	[50]
1089	161	1288	40	5	1	≥1	-		[Han]
1090	81	648	40	5	2	≥1	-	2#243	
1091	41	328	40	5	4	≥1	-	4#45,D#1193*	
1092	33	264	40	5	5	≥1	-		[Han]
1093	21	168	40	5	8	$\geq 10^{24}$	-	8#6	[50]
1094	17	136	40	5	10	≥1	-	2#246	
1095	11	88	40	5	16	$\geq 10^{7}$	-	8#7	[50]

No	v	b	r	k	λ	Nd	Nr	Comments	References
1096	201	1340	40	6	1	?	-		
1097	51	340	40	6	4	≥1	-	4#48*,2#248	
1098	21	140	40	6	10	≥1	-	5#25*,2#249	
1099	49	280	40	7	5	$\geq 10^{53}$	$\geq 10^{53}$	5#26	[50]
1100	21	120	40	7	12	≥1	?	4#49	
1101	281	1405	40	8	1	?	-		
1102	141	705	40	8	2	?	-		
1103	71	355	40	8	4	≥1	-		[W]
1104	57	285	40	8	5	$\geq 10^{63}$	-	5#27	[50]
1105	41	205	40	8	7	≥1	-	D#1194*	[Han]
1106	36	180	40	8	8	?	-		
1107	29	145	40	8	10	≥1	-	4#50*,2#251*	[Han]
1108	21	105	40	8	14	≥1	-	5#28*	[Han]
1109	81	360	40	9	4	$\geq 10^{108}$	$\geq 10^{108}$	4#51	[50]
1110	361	1444	40	10	1	?	-		
1111	181	724	40	10	2	?	-	2#253*	
1112	121	484	40	10	3	?	-		
1113	91	364	40	10	4	$\geq 10^{125}$	-	4#52	[50]
1114	73	292	40	10	5	≥1	-		[W]
1115	61	244	40	10	6	?	-	2#255*	
1116	46	184	40	10	8	?	-	4#53*,2#256*	
1117	41	164	40	10	9	≥1	-	D#1195*	[Han]
1118	37	148	40	10	10	≥1	-	2#257	
1119	31	124	40	10	12	≥1	-	4#54	
1120	25	100	40	10	15	≥1	-		[Han]
1121	21	84	40	10	18	≥1	-	2#259	
1122	441	1470	40	12	1	?	-		
1123	111	370	40	12	4	?	-	2#260*	
1124	45	150	40	12	10	?	-	2#261*	
1125	481	1480	40	13	1	?	?		
1126	105	300	40	14	5	?	-		
1127	561	1496	40	15	1	?	-		
1128	141	376	40	15	4	?	-	2#262*	
1129	81	216	40	15	7	≥1	-		[W]
1130	57	152	40	15	10	?	-	2#263*	
1131	36	96	40	15	16	≥1	-	2#264	
1132	76	190	40	16	8	≥1	-	2#265	
1133	171	380	40	18	4	?	-	2#266*	
1134	361	760	40	19	2	≥1	≥1	2#267	
1135	761	1522	40	20	1	?	-		
1136	381	762	40	20	2	≥1	-	2#268	
1137	191	382	40	20	4	?	-	2#269*	
1138	153	306	40	20	5	?	-		
1139	96	192	40	20	8	≥1	-	2#270	
1140	77	154	40	20	10	?	-	2#271*	
1141	41	82	40	20	19	≥1	-	D#1196	[74]
1142	121	220	40	22	7	?	-		
1143	921	1535	40	24	1	?	-		
1144	231	385	40	24	4	?	-		
1145	93	155	40	24	10	?	-		

No	v	b	r	k	λ	Nd	Nr	Comments	References
1146	65	104	40	25	15	?	-	R#1171*	
1147	1001	1540	40	26	1	?	-		
1148	81	120	40	27	13	$\geq 10^{13}$	$\geq 10^{13}$	R#1170,AG	[H,32,79]
1149	217	310	40	28	5	?	-		
1150	91	130	40	28	12	?	-	R#1169*	
1151	1161	1548	40	30	1	?	-		
1152	291	388	40	30	4	?	-		
1153	117	156	40	30	10	?	-	R#1168*	
1154	156	195	40	32	8	?	-	R#1167*	
1155	221	260	40	34	6	?	-	R#1166*	
1156	273	312	40	35	5	?	-	R#1165*	
1157	351	390	40	36	4	?	-	R#1164*	
1158	481	520	40	37	3	?	0	R#1163*,NE3	[K]
1159	741	780	40	38	2	0	-	R#1162*,NE2	[H]
1160	1521	1560	40	39	1	?	?	R#1161*,AG	
1161	1561	1561	40	40	1	?	-	PG	
1162	781	781	40	40	2	0	-	NE1	[H]
1163	521	521	40	40	3	?	-		
1164	391	391	40	40	4	?	-		
1165	313	313	40	40	5	0	-	NE1	[H]
1166	261	261	40	40	6	0	-	NE1	[H]
1167	196	196	40	40	8	0	-	NE1	[H]
1168	157	157	40	40	10	0	-	NE1	[H]
1169	131	131	40	40	12	?	-		
1170	121	121	40	40	13	$\geq 10^{29}$	-	PG	[H,32,79]
1171	105	105	40	40	15	?	-		
1172	42	574	41	3	2	$\geq 6 \times 10^{24}$	≥ 1	D	[Han,33,50]
1173	124	1271	41	4	1	≥ 2	≥ 1		[Han,11,35]
1174	165	1353	41	5	1	≥ 1	?		[Han]
1175	42	287	41	6	5	≥ 1	?	D	[Han]
1176	42	246	41	7	6	≥ 1	?	D	[Han]
1177	288	1476	41	8	1	?	?		
1178	370	1517	41	10	1	?	?		
1179	247	779	41	13	2	?	?		
1180	42	123	41	14	13	?	?	D	
1181	247	533	41	19	3	?	?		
1182	780	1599	41	20	1	?	?		
1183	42	82	41	21	20	≥ 1	0	R#1196,NE3	[K,74]
1184	124	164	41	31	10	?	0	R#1195*,NE3	[K]
1185	165	205	41	33	8	?	0	R#1194*,NE3	[K]
1186	288	328	41	36	5	?	0	R#1193*,NE3	[K]
1187	370	410	41	37	4	?	0	R#1192*,NE3	[K]
1188	780	820	41	39	2	?	0	R#1191*,NE3	[K]
1189	1600	1640	41	40	1	?	?	R#1190*,AG	
1190	1641	1641	41	41	1	?	-	PG	
1191	821	821	41	41	2	?	-		
1192	411	411	41	41	4	?	-		
1193	329	329	41	41	5	?	-		
1194	206	206	41	41	8	0	-	NE1	[H]
1195	165	165	41	41	10	?	-		
1196	83	83	41	41	20	≥ 1	-		[74]

References

[H] M. Hall Jr., *Combinatorial Theory*, Blaisdell, Waltham, Mass. 1967.

[Han]
H. Hanani, Balanced incomplete block designs and related designs, *Discrete Mathematics* 11 (1975), 255-369.

[K] S. Kageyama, A survey of resolvable solutions of balanced incomplete block designs, *Rev. Inst. Internat. Stat.* 40 (1972), 269-273.

[T] K. Takeuchi, A table of difference sets generating balanced incomplete block designs, *Rev. Inst. Internat. Stat.* 30 (1962), 361-366.

[W] R.M. Wilson, Cyclotomy and difference families in elementary abelian groups, *Journal of Number Theory* 4 (1972), 17-47.

[D] J.W. DiPaola, J.S. Wallis, W.D. Wallis, A list of (v,b,r,k,λ) designs for $r \leq 30$, *Proc. 4th S.-E. Confer. Combinatorics, Graph Theory and Computing* 1973, 249-258.

[1] M. Aschbacher, On collineation groups of symmetric block designs, *J. Combinatorial Theory (A)* 11 (1971), 272-281.

[2] E.F. Assmus Jr., J.A. Mezzaroba, C.J. Salwach, Planes and biplanes, *Higher Combinatorics* (ed. M. Aigner), D. Reidel, Dordrecht 1977, 205-212.

[3] R.D. Baker, Resolvable BIBDs and SOLS, *Discrete Mathematics* 44 (1983), 13-29.

[4] H. Beker, F. Piper, Some designs which admit strong tactical decompositions, *J. Combinatorial Theory (A)* 22 (1977), 38-42.

[5] C.T. Benson, Two nonisomorphic (v,k,λ) designs with similar incidence matrices, *J. Combinatorial Theory* 5 (1968), 419-421.

[6] V.N. Bhat, Nonisomorphic solutions of some BIBDs, III, *J. Combinatorial Theory (A)* 12 (1972), 225-252.

[7] V.N. Bhat, S.S. Shrikhande, Nonisomorphic solutions of some balanced incomplete block designs I., *J. Combinatorial Theory* 9 (1970), 174-191.

[8] V.N. Bhat-Nayak, V.D. Kane, W.L. Kocay, R.G. Stanton, Settling some BIBD conjectures, *Ars Combinatoria* 16 (1983), 229-234.

[9] D. Breach, A.R. Thompson, Reducible 2-(11,5,4) and 3-(12,6,4) designs, Research Report 31/1983, Dept. of Math., University of Canterbury, Christchurch, N.Z.

[10] W.G. Bridges, M. Hall Jr., J.L. Hayden, Codes and designs, *J. Combinatorial Theory (A)* 31 (1981), 155-174.

[11] A.E. Brouwer, Some nonisomorphic BIBDs (4,1;v), Math. Centrum Amsterdam, ZW 102 (1977).

[12] A.E. Brouwer, Some unitals on 28 points and their embeddings in projective planes of order 9, *Geometries and Groups*, Proc. Colloq. Berlin 1981, Lecture Notes Math. 893, Springer 1981, 183-188.

[13] A.E. Brouwer, Private communication.

[14] A.E. Brouwer, H.A. Wilbrink, A symmetric design with parameters 2-(49,16,5), *J. Combinatorial Theory (A)* (to appear).

[15] F.C. Bussemaker, R. Mathon, J.J. Seidel, Tables of two-graphs, *Combinatorics and Graph Theory*, Proc. Conf. Calcutta 1980, Lecture Notes Math. 885, Springer 1981, 70-112.

[16] W.H. Clatworthy, R.J. Lewyckyj, Comments on Takeuchi's table of difference sets generating balanced incomplete block designs, *Rev. Inst. Internat. Stat.* 36 (1968), 12-18.

[17] C.J. Colbourn, M.J. Colbourn, On cyclic Steiner systems S(2,6,91), *Abstracts Amer. Math. Soc.* 2 (1981), 463.

[18] C.J. Colbourn, M.J. Colbourn, J.J. Harms, A. Rosa, A complete census of (10,3,2) block designs and of Mendelsohn triple systems of order ten, III. (10,3,2) block designs without repeated blocks, *Congr. Numer.* 39 (1983), 211-234.

[19] M.J. Colbourn, Cyclic Block Designs: Computational Aspects of Their Construction and Analysis, Dept. of Computer Science, Univ. of Toronto, Technical Report 146/80.

[20] R.L. Constable, Some nonisomorphic solutions for the BIBD(12,33,11,4,3), *Utilitas Math.* 15 (1979), 323-333.

[21] D.G. Corneil, R.A. Mathon, Algorithmic techniques for the generation and analysis of strongly regular graphs and other combinatorial configurations, *Annals of Discrete Mathematics* 2 (1978), 1-32.

[22] P. Dembowski, *Finite Geometries*, Springer, N.Y. 1968.

[23] R.H.F. Denniston, On biplanes with 56 points, *Ars Combinatoria* 9 (1980), 167-179.

[24] R.H.F. Denniston, A Steiner system with a maximal arc, *Ars Combinatoria* 9 (1980), 247-248.

[25] R.H.F. Denniston, Enumeration of symmetric designs (25,9,3), *Annals of Discrete Mathematics* 15 (1982), 111-127.

[26] R.A. Fischer, F. Yates, *Statistical Tables for Biological, Agricultural and Medical Research*, 6th ed., Hafner, New York 1963.

[27] B. Ganter, A Gulzow, R. Mathon, A. Rosa, A complete census of (10,3,2)-block designs and of Mendelsohn triple systems of order ten, IV. (10,3,2)-designs with repeated blocks, *Math. Schriften Kassel* 5/78.

[28] P. Gibbons, Computing techniques for the construction and analysis of block designs, Dept. of Computer Science, Univ. of Toronto, Technical Report 92/76.

[29] H.-D.O.F. Gronau, A survey of results on the number of t-(v,k,λ)-designs (to appear).

[30] H.-D.O.F. Gronau, J. Prestin, Some results on designs with repeated blocks, *Rostock. Math. Kolloq.* 20 (1982), 15-38.

[31] W. Haemers, Eigenvalue techniques in design and graph theory (thesis), Tract 121, Math. Centrum Amsterdam 1980.

[32] M. Hall, Jr., A survey of difference sets, *Proc. Amer. Math. Soc.* 7 (1956), 975-986.

[33] H. Hanani, On resolvable balanced incomplete block designs, *J. Combinatorial Theory (A)* 17 (1974), 275-289.

[34] H. Hanani, Unpublished note.

[35] H. Hanani, D.K. Ray-Chaudhuri, R.M. Wilson, On resolvable designs, *Discrete Mathematics* 3 (1972), 343-357.

[36] N. Ito, J.S. Leon, J.Q. Longyear, Classification of 3-(24,12,5) designs and 24-dimensional Hadamard matrices, *J. Combinatorial Theory (A)* 31 (1981), 66-93.

[37] A.V. Ivanov, Constructive enumeration of incidence systems, *Annals of Discrete Mathematics* (to appear).

[38] Z. Janko, Tran Van Trung, On projective planes of order 12 with an automorphism of order 13, I, *Geom. Dedicata* 11 (1981), 257-284.

[39] P.W.M. John, A balanced design for 18 varieties, *Technometrics* 15 (1973), 641-642.

[40] N.L. Johnson, The translation planes of order 16 that admit nonsolvable collineation groups, *Math. Zeitschr.* (to appear).

[41] N.L. Johnson, Private communication.

[42] S. Kageyama, A resolvable solution of BIBD(18,51,17,6,5), *Ars Combinatoria* 15 (1983), 315-316.

[43] W. Kantor, Private communication.

[44] C.C. Lindner, A. Rosa, On the existence of automorphism free Steiner triple systems, *J. Algebra* 34 (1975), 430-443.

[45] J.H. van Lint, H.C.A. van Tilborg, J.R. Wichsmar, Block designs with $v = 10$, $k = 5$, $\lambda = 4$, *J. Combinatorial Theory (A)* 23 (1977), 105-115.

[46] R. Mathon, Constructions for cyclic Steiner 2-designs (to appear).

[47] R. Mathon, K.T. Phelps, A. Rosa, Small Steiner triple systems and their properties, *Ars Combinatoria* 15 (1983), 3-110.

[48] R. Mathon, A. Rosa, A census of Mendelsohn triple systems of order nine, *Ars Combinatoria* 4 (1977), 309-315.

[49] R. Mathon, A. Rosa, The 4-rotational Steiner and Kirkman triple systems of order 21, *Ars Combinatoria* 17A (1984), 241-250.

[50] R. Mathon, A. Rosa, Some results on the existence and enumeration of BIBDs, Report (to appear).

[51] R.B. McFeat, A. Neumaier, Some tuple systems constructions with applications to resolvable designs, *J. Combinatorial Theory (A)* 36 (1984), 92-100.

[52] B.D. McKay, R.G. Stanton, Isomorphism of two large designs, *Ars Combinatoria* 6 (1978), 87-90.

[53] W.H. Mills, A new block design, *Proc. 6th S.-E. Conf. Combinatorics, Graph Theory and Computing*, Boca Raton 1975, 461-465.

[54] W.H. Mills, Two new block designs, *Utilitas Math.* 7 (1975), 73-75.

[55] W.H. Mills, The construction of balanced incomplete block designs with $\lambda = 1$, *Proc. 7th Manitoba Conf. Numerical Math. and Computing*, Winnipeg 1977, 131-148.

[56] W.H. Mills, The construction of BIBDs using nonabelian groups, *Proc. 9th S.-E. Conf. Combinatorics, Graph Theory and Computing*, Boca Raton 1978, 519-526.

[57] W.H. Mills, The construction of balanced incomplete block designs, *Proc. 10th S.-E. Conf. Combinatorics, Graph Theory and Computing*, Boca Raton 1979, vol.I, 73-86.

[58] E.J. Morgan, Some small quasi-multiple designs, *Ars Combinatoria* 3 (1977), 233-250.

[59] H.K. Nandi, Enumeration of nonisomorphic solutions of balanced incomplete block designs, *Sankhya* 7 (1946), 305-312.

[60] C.W. Norman, Nonisomorphic Hadamard designs, *J. Combinatorial Theory (A)* 21 (1976), 336-344.

[61] S.C. Panandikar, On a family of resolvable BIB designs, *Calcutta Statist. Assoc. Bull.* 31 (1981), 123-128.

[62] D.A. Preece, Incomplete block designs with v = 2k, *Sankhya (A)* 29 (1967), 305-316.

[63] D.P. Rajkundlia, Some techniques for constructing infinite families of BIBDs, *Discrete Mathematics* 44 (1983), 64-96.

[64] B.T. Rumov, The existence of some resolvable block designs with divisibility into groups, *Math. Notes* 19 (1976), 376-382.

[65] E. Seiden, A method of construction of resolvable BIBD, *Sankhya (A)* 25 (1963), 393-394.

[66] S.S. Shrikhande, On a two-parameter family of balanced incomplete block designs, *Sankhya (A)* 24 (1962), 38-40.

[67] S. Sillito, An extension property of a class of balanced incomplete block designs, *Biometrika* 44 (1957), 278-279.

[68] R.G. Stanton, R.C. Mullin, Uniqueness theorems in balanced incomplete block designs, *J. Combinatorial Theory* 7 (1969), 37-48.

[69] D.R. Stinson, H. Ferch, 2000000 Steiner triple systems of order 19 (to appear).

[70] D.R. Stinson, S.A. Vanstone, Some non-isomorphic Kirkman triple systems of order 39 and 51 (to appear).

[71] V.D. Tonchev, Hadamard matrices of order 28 with automorphism of order 13, *J. Combinatorial Theory (A)* 35 (1983), 43-57.

[72] V.D. Tonchev, R.V. Raev, Cyclic 2-(13,5,5) designs, *C.R. Acad. Bulg. Sci.* 35 (1982), 1205-1208.

[73] Tran Van Trung, The existence of symmetric block designs with parameters (41,16,6) and (66,26,10), *J. Combinatorial Theory (A)* 33 (1982), 201-204.

[74] W.D. Wallis, A.P. Street, J.S. Wallis, *Combinatorics: Room Squares, Sum-Free Sets Hadamard Matrices*, Lecture Notes Math. 292, Springer, Berlin 1972.

[75] R.M. Wilson, Nonisomorphic Steiner triple systems, *Math. Zeitschr.* 135 (1974), 303-313.

[76] R.M. Wilson, Symmetric group-divisible designs, Manuscript.

[77] R.H.F. Denniston, Some maximal arcs in finite projective planes, *Journal of Combinatorial Theory* 6 (1969), 317-319.

[78] Z. Janko, Tran Van Trung, Construction of new symmetric block designs (to appear).

[79] D. Jungnickel, The number of designs with classical parameters grows exponentially, *Geom. Dedicata* 16 (1984) 167-178.

[80] D. Jungnickel, Quasimultiples of biplanes and residual biplanes, *Ars Combinatoria* (to appear).

[81] A.Y. Petrenyuk, O postroenii neizomorfnykh blok-skhem s pomošćiu izografičeskikh naborov blokov, Kirovogr. Inst. S.-Kh. Mašinostr., Kirovograd 1983, 16 pp.

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 309–320

On the Existence of Strong Kirkman Cubes of Order 39 and Block Size 3

A. Rosa

Department of Mathematics
McMaster University
Hamilton, Ontario
CANADA

S.A. Vanstone

Department of Combinatorics and Optimization
St. Jerome's College
University of Waterloo
Waterloo, Ontario
CANADA

Abstract

In this article we are interested in resolvable $(v,k,1)$-BIBDs. Let D be a $(v,k,1)$-BIBD and let R and S be two resolutions of the blocks of D. R and S are said to be orthogonal resolutions if any parallel class from R has at most one block in common with any class of S. A set of t resolutions of D is called a set of t orthogonal resolutions if every pair of these resolutions is orthogonal. For $t = 2$ the design resulting is called a Kirkman square. For $t = 3$ it is called a strong Kirkman cube. Previously, the smallest order for which a strong Kirkman cube of block size 3 was known to exist was $v = 255$. This paper gives an algorithm for searching for a particular type of Kirkman square with block size 3. The algorithm was applied to the case $v = 39$, $k = 3$ with the result that several strong Kirkman cubes were found. The designs obtained have automorphism groups which are transitive on parallel classes of all three orthogonal resolutions. In order to find the strong Kirkman cubes of order 39 cited above we enumerate all Kirkman squares of order 39 of a specific type.

1. Introduction.

This paper deals with resolvable $(v,k,1)$-BIBDs. In particular, $v = 39$, $k = 3$ is our main concern. Recall that a $(v,k,1)$-BIBD is resolvable if its blocks can be partitioned into classes (called parallel classes) $R_1, R_2, \ldots, R_r$ ($r = \frac{v-1}{k-1}$) such that each element is contained in a unique block of each class. The set R = $\{R_i\colon 1 \le i \le r\}$ is called a resolution of D. A $(v,k,1)$-BIBD with a resolution R is called a Kirkman system. Let R = $\{R_i\colon 1 \le i \le r\}$ and S = $\{S_i\colon 1 \le i \le r\}$ be two resolutions of the same $(v,k,1)$-BIBD. R is said to be orthogonal to S provided

$$|R_i \cap S_j| \le 1, \quad 1 \le i,\ j \le r.$$

A set of t resolutions of a $(v,k,1)$-BIBD is called a set of t orthogonal resolutions if every pair of distinct resolutions in the set is orthogonal. Although the sequel does not pertain to it, an interesting question is to determine good upper and lower bounds on t for given v and k.

A $(v,k,1)$-BIBD with t orthogonal resolutions is called a Kirkman square when $t = 2$ and a strong Kirkman cube when $t = 3$. We denote these by $KS_k(v)$ and $SKC_k(v)$ respectively. For the definition of a weak Kirkman cube the reader is referred to [8].

A necessary condition for the existence of $KS_k(v)$ and $SKC_k(v)$ is $v \equiv k(\mathrm{mod}\, k(k-1))$. In the case $k = 2$ the existence question is completely settled by the following two theorems.

Theorem 1.1 ([7]) For each positive integer $v \equiv 0 \pmod 2$, $v \ne 4$ or 6, there exists a $KS_2(v)$. There does not exist a $KS_2(4)$ or a $KS_2(6)$.

Theorem 1.2 ([2]) For each positive integer $v \equiv 0 \pmod 2$, $v \ne 4$ or 6, there exists an $SKC_2(v)$. There does not exist an $SKC_2(4)$ or an $SKC_2(6)$.

The existence question for $k \ge 3$ remains open. In the case $k = 3$ an asymptotic existence result can be stated.

Theorem 1.3 ([9]) There exists a constant v_1 such that for all $v > v_1$ and $v \equiv 3 \pmod 6$ there exists a $KS_3(v)$.

This paper is only concerned with the case $k = 3$; in this case, the underlying design is a Steiner triple system (STS). At present there are very few direct constructions for $KS_3(v)$ and $SKC_3(v)$ and until recently the only $SKC_3(v)$ constructed directly had order $v = 255$. This article examines a particular class of such designs for $v = 39$ with the hope that a complete enumeration may shed some light on a more general direct construction. The choice of $v = 39$ is easily explained. We are interested in finding direct constructions for $KS_3(v)$s with automorphism groups which are cyclic on the parallel classes. Such $KS_3(v)$'s we will call cyclic or starter-adder. This requirement restricts our attention to $v \equiv 3 \pmod{12}$. It is not difficult to establish that no such design exists for $v = 15$. In fact, there is no $KS_3(15)$ whatsoever. This can be seen by exhaustively examining all resolvable (15,3,1)-BIBDs (for a listing, see, e.g. [6]). A $KS_3(27)$ was found in ([5]) and, more recently, a cyclic $KS_3(27)$ was found [4]. In fact, Janko and Van Trung ([4]) have done a complete enumeration of all such cyclic designs of order 27 and found precisely three nonisomorphic $KS_3(27)$s. The next case to consider for cyclic systems is $v = 39$. In the next section we describe the class of cyclic systems which we intend to enumerate.

2. A special class of Kirkman systems.

In order to construct $KS_3(39)$s we need to construct Kirkman triple systems. Since we are interested only in cyclic $KS_3(39)$s we can assume that the set of elements of such a design is $V = \mathbf{Z}_{19} \times \{0,1\} \cup \{\infty\}$, and the corresponding cyclic automorphism is $\alpha = (0_1 1_1 \ldots 18_1)(0_2 1_2 \cdots 18_2)(\infty)$. A convenient way to represent one of these designs is to list the blocks of one parallel class from each of the two orthogonal resolutions. Suppose $\{B_1, B_2, \ldots, B_r\}$ and $\{C_1, C_2, \ldots, C_r\}$ are parallel classes of blocks, one from each of the orthogonal resolutions. Since $v = 39$ and $r = 19$, the blocks of the underlying design fall into orbits of length 19 under the action of the group. Hence, we can assume that B_i and C_i are in the same orbit, $1 \le i \le r$. Therefore, an alternate way to list the cyclic $KS_3(39)$ is to list $\{B_1, B_2, \ldots, B_r\}$ and a set of mappings $\{a_1, a_2, \ldots, a_r\}$ such that $a_i(B_i) = C_i$, $1 \le i \le r$. This method of listing is commonly referred to as starter-adder. The set $\{B_1, B_2, \ldots, B_r\}$ is the starter and the set $\{a_1, a_2, \ldots, a_r\}$ is the adder. As an example consider one of the cyclic $KS_3(27)$'s found in [4].

$B_1 = \{\infty\ 0_1\ 0_2\}$	$B_6 = \{8_1\ 4_1\ 6_0\}$
$B_2 = \{7_0\ 3_0\ 11_1\}$	$B_7 = \{9_1\ 3_1\ 4_0\}$
$B_3 = \{10_0\ 5_0\ 6_1\}$	$B_8 = \{11_0\ 12_0\ 1_0\}$
$B_4 = \{2_0\ 9_0\ 12_1\}$	$B_9 = \{5_1\ 7_1\ 10_1\}$
$B_5 = \{2_1\ 1_1\ 8_0\}$	

$a_1 = 0$, $a_2 = 1$, $a_3 = 5$, $a_4 = 3$, $a_5 = 6$, $a_6 = 10$, $a_7 = 7$, $a_8 = 8$, $a_9 = 12$ where $a_i(\{a_h, b_l, c_s\}) = \{(a+a_i)_h, (b+a_i)_l, (c+a_i)_s\}$ and operations are in the integers modulo 13.

Adopting the notation of the previous paragraph, we will specify a cyclic $KS_3(39)$ by a "base" set S of triples (13 of them) on the symbols $V = \mathbf{Z}_{19} \times \{0,1\} \cup \{\infty\}$ and an adder set A of 13 elements from $\mathbf{Z}_{19}$. We note that the adder A must consist of 13 distinct elements.

Before continuing with the description of the class of cyclic $KS_3(39)$ to be investigated we require several more definitions.

Let D be a $(v,k,1)$-BIBD with element set V and block set B. A $(u,k,1)$-BIBD D' with element set V' and block set B$'$ is a subdesign of D if $V' \subseteq V$ and B$' \subseteq$ B.

Let G be a finite abelian group of odd order $2n+1$ which is written additively. A strong starter T in G is a partition of the nonzero elements of G into pairs $\{x_1,y_1\}, \ldots, \{x_n,y_n\}$ such that

(i) $\bigcup_{i=1}^{n} \{\pm(x_i-y_i)\} = G\backslash\{0\}$,

(ii) $x_i + y_i \neq 0, \; i = 1,2,\ldots,n$

(iii) $x_i + y_i \neq x_j + y_j, \; i,j = 1,2,\ldots,n, \; i \neq j$.

Consider a cyclic $KS_3(v)$ D defined on the set $V = \mathbf{Z}_r \times \{0,1\} \cup \{\infty\}$ where $r = \frac{v-1}{2}$. Let $S = \{B_0,B_1, \ldots ,B_m\}$, where $m = \frac{v-3}{3}$, be a starting set of blocks. Without loss of generality assume that $B_0 = \{\infty\ 0_0\ 0_1\}$. Suppose D contains a subdesign D' of order r which is fixed by the automorphism group. Without loss of generality we can assume that D' is defined on the point set $\mathbf{Z}_r \times \{1\}$ and that $B_h,B_{h+1}, \ldots ,B_m$ is a base set of triples for D' ($h = \frac{r+1}{2}$). Let $B_i = \{a_0^i,b_0^i,c_1^i\}$, $1 \leq i \leq h-1$. If $T = \{\{a_0^i,b_0^i\}\colon 1 \leq i \leq h-1\}$ is a strong starter in $\mathbf{Z}_r$ and $c^i = \frac{a^i+b^i}{2}$ then D is called a $KS_3^*(v)$. In the next two sections we will enumerate all $KS_3^*(v)$'s for $v = 27$ and 39.

3. The Algorithm.

The algorithm we use for enumerating $KS_3^*(v)$ requires the following concept. Any partition of the set $\{1,2,\ldots,3t\}$ into triples such that in each triple the sum of two of the numbers is equal to the third or the sum of the three numbers is equal to $6t+1$ is called a solution to the first Heffter's difference problem for t. We denote this problem by $HDP(t)$. Heffter ([3]) observed that any solution to $HDP(t)$ can be used to construct a cyclic Steiner triple system (STS) of order $6t+1$. A cyclic STS of order v is a $(v,3,1)$-BIBD that has an automorphism consisting of a single cycle of length v. Without loss of generality one can assume that the point set is $V = \mathbf{Z}_v$ and the cyclic automorphism is $i \to i+1 \pmod v$. If $\{\{a_i,b_i,c_i\}\colon 1 \leq i \leq t\}$ is a solution to

$HDP(t)$ then the set of base triples $\{\{0,a_i,a_i+b_i\}: 1 \le i \le t\}$ generates an STS of order $6t+1$. Conversely, let (V,B) be a cyclic STS of order $6t+1$. For $x,y \in V$ define

$$\Delta(x,y) = \min(|x-y|, 6t+1-|x-y|)$$

and for $\{x,y,z\} \in \mathrm{B}$ let

$$\Delta\{x,y,z\} = \{\Delta(x,y),\Delta(x,z),\Delta(y,z)\}.$$

Then $\{\Delta B: B \in \mathrm{B}\}$ is a solution to $HDP(t)$.

There is precisely one solution to $HDP(2)$ and four solutions to $HDP(3)$. We list these for later reference.

$$HDP(2): \begin{matrix}\{134\}\\ \{256\}\end{matrix}$$

$$HDP(3): \begin{matrix}\{178\} & \{134\} & \{145\} & \{156\}\\ \{235\} & \{279\} & \{268\} & \{289\}\\ \{469\} & \{568\} & \{379\} & \{347\}\end{matrix}$$

For a more extensive listing of solutions to $HDP(t)$ the reader is referred to [1].

Let $S = \{\{x_i,y_i\}: 1 \le i \le \frac{r-1}{2}\}$ be a strong starter in the cyclic group $\mathbf{Z}_r$, $r = 6t+1$. Let $\bar{S} = \mathbf{Z}_r \setminus \{(x_i+y_i)/2: 1 \le i \le \frac{r-1}{2}\}$. Hence, $|\bar{S}| = 3t$. Since $x_i + y_i \ne x_j + y_j$, $i \ne j$ (S is a strong starter) then $(x_i+y_i)/2 \ne (x_j+y_j)/2$ for $i \ne j$. S is said to be a proper strong starter if there exists a partition $P(\bar{S})$ of $\bar{S}$ into t triples

$$P(\bar{S}) = \{\{u_i,v_i,w_i\}: 1 \le i \le t\}$$

such that $\{\Delta\{u_i,v_i,w_i\}: 1 \le i \le t\}$ is a solution to $HDP(t)$.

If S is a proper strong starter then it is a simple matter to construct a cyclic resolvable $(12t+3,3,1)$-BIBD D on the set $\mathbf{Z}_r \times \{0,1\} \cup \{\infty\}$ where D contains a subdesign of order r fixed by the cyclic automorphism.

We note that if $\{a,b,c\}$, $a < b < c$ is a triple in a solution to $HDP(t)$ then the corresponding base triple in the cyclic $STS(6t+1)$ can be taken to be either $\{0,a,a+b\}$ or $\{0,-a,-(a+b)\}$. We label the first block "+" and the second "-". If the triples in a solution to $HDP(t)$ are arbitrarily ordered then there are 2^t ways to construct the base blocks for an STS and each can be labelled with a sequence of length t consisting of "+" and "-"s where a $+(-)$ in the ith positions indicate that the ith triple in the solution was replaced by a $+(-)$ difference block. If S is a proper strong starter then there is at least one of these 2^t base sets which can be translated to cover all elements in $\bar{S}$. For the purposes of our algorithm we determine all ways to partition $\bar{S}$ and label each partitioning with one of the 2^t possibilities. Having found all proper strong starters and all possible partitionings, for each we check the resulting base resolutions for orthogonality. We only check two base resolutions coming from the strong starters S and S' if the partitionings of $\tilde{S}$ and $\tilde{S}'$ have the same label or the labels are negatives of each other. (Two labels are negatives of each other if one can be obtained from the other by interchanging "+" and "-" signs.)

4. Results for $v = 27$ and $v = 39$.

Let $S = \{\{x_i, y_i\}: 1 \leq i \leq t\}$ be a strong starter in $\mathbf{Z}_r$. Define

$$aS = \{\{ax_i, ay_i\}: 1 \leq i \leq t\}.$$

aS is a strong starter iff a^{-1} exists. Two strong starters S and S' are said to be equivalent if there exists $a \in \mathbf{Z}_r$ such that $S' = aS$.

There are precisely 2 inequivalent strong starters in $\mathbf{Z}_{13}$. They are:

$$S = \{\{2,4\}, \{3,7\}, \{6,12\}, \{5,10\}, \{1,11\}, \{8,9\}\}$$

$$S' = \{\{5,7\}, \{3,12\}, \{2,8\}, \{6,11\}, \{1,4\}, \{9,10\}\}$$

$$\tilde{S} = \tilde{S}' = \{4,7,8,10,11,12\}.$$

There is precisely one solution to $HDP(2)$ and a simple check shows that both S and S' are proper. The resulting base resolutions are not orthogonal and so no $KS_3^*(27)$ exists. Of course, this result follows immediately from the fact that Janko and Van Trung [1] enumerated all $KS_3(27)$ which admit a cyclic automorphism of order 13 and it can be easily checked that none of these fixes a subdesign of order 13. We only include the result on $KS_3^*(27)$s for completeness.

The results for $v = 39$ are more encouraging. In $\mathbf{Z}_{19}$ there are precisely 51 non-equivalent strong starters. As observed earlier there are 4 solutions to $HDP(3)$. The algorithm produced 49 non-equivalent $KS_3^*(39)$s. By "non-equivalent" we mean that one cannot be obtained from the other by multiplying by some element in $\mathbf{Z}_{19}$. All systems are displayed in table 1. Solutions are displayed as a base parallel class and associated adders. For example, the first few entries in the table are:

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

The first line is a base parallel class which for conciseness is written as a strong starter and a base set of triples for the subdesign of order 19. The triple $\{\infty\ 0_0\ 0_1\}$ is omitted. In its expanded form the first line would read:

$$\{3_0\ 5_0\ 4_1\}, \{11_0\ 15_0\ 13_1\}, \{7_0\ 13_0\ 10_1\}, \{4_0\ 12_0\ 8_1\},$$

$$\{1_0\ 10_0\ 15_1\}, \{6_0\ 18_0\ 12_1\}, \{2_0\ 16_0\ 9_1\}, \{14_0\ 17_0\ 6_1\},$$

$$\{8_0\ 9_0\ 18_1\}, \{2_1\ 1_1\ 17_1\}, \{5_1\ 7_1\ 14_1\}, \{3_1\ 11_1\ 16_1\}, \{\infty\ 0_0\ 0_1\}.$$

The adder elements are added to the corresponding triples to produce a base parallel class for an orthogonal resolution. In this case the parallel class is:

$$\{14_0,16_0,15_1\},\ \{6_0,10_0,8_1\},\ \{9_0,15_0,12_1\},\ \{12_0,1_0,16_1\}.$$

$$\{4_0,13_0,18_1\},\{18_0,11_0,5_1\},\ \{3_0,17_0,10_1\},\ \{2_0,5,13_1\}.$$

$$\{7_0,8_0,17_1\},\ \{7_1,6_1,3_1\},\ \{2_1,4_1,11_1\},\ \{1_1,9_1,14_1\},\ \{\infty,0_0,0_1\}.$$

5. Strong Kirkman Cubes.

The smallest order for which a strong Kirkman cube was known to exist is 255. This design is cyclic. That is, all three orthogonal resolutions are generated by a cyclic automorphism of order 127. It is not known whether an $SKC_3(27)$ exists. A simple check of the Janko-Van Trung paper [4] shows that there is no cyclic $SKC_3(27)$. By examining the $KS_3^*(39)$s of Table 1 we find precisely 2 non-equivalent cubes. These cubes are displayed below as a base parallel class and two adders from which one gets two other base parallel classes for the orthogonal resolutions.

$\{8_0\ 6_0\ 7_1\}$	$\{1_0\ 5_0\ 3_1\}$	$\{11_0\ 17_0\ 14_1\}$	$\{2_0\ 10_0\ 6_1\}$	$\{7_0\ 16_0\ 2_1\}$	$\{3_0\ 15_0\ 9_1\}$
10	1	11	5	7	17
11	14	2	18	3	8

$\{4_0\ 18_0\ 11_1\}$	$\{9_0\ 12_0\ 1_1\}$	$\{13_0\ 14_0\ 4_1\}$	$\{12_1\ 13_1\ 16_1\}$	$\{8_1\ 15_1\ 17_1\}$	$\{5_1\ 10_1\ 18_1\}$
13	15	16	2	14	3
1	7	12	6	4	9

CUBE 1.

$\{6_0\ 8_0\ 7_1\}$	$\{3_0\ 7_0\ 5_1\}$	$\{1_0\ 14_0\ 17_1\}$	$\{5_0\ 13_0\ 9_1\}$	$\{2_0\ 11_0\ 16_1\}$	$\{10_0\ 17_0\ 4_1\}$
10	3	14	17	2	16
8	6	9	7	4	1

$\{4_0\ 18_0\ 11_1\}$	$\{9_0\ 12_0\ 1_1\}$	$\{15_0\ 16_0\ 6_1\}$	$\{3_1\ 15_1\ 18_1\}$	$\{2_1\ 10_1\ 12_1\}$	$\{8_1\ 13_1\ 14_1\}$
13	15	5	7	11	1
18	12	11	3	2	14

CUBE 2.

These cubes are the smallest examples of strong Kirkman cubes that we are aware of. (For various recursive constructions that produce infinite families from these designs, see [9].) The only other order that we know for which a cube is constructed directly, is $v = 255$ (cf. above). It is constructed from the points and lines of $PG(7,2)$ and all three orthogonal resolutions can be cyclically generated. Also, the automorphism fixes a subdesign of order 127.

An exhaustive check shows that neither of the two cubes $SKC_3(39)$ found can be extended cyclically to yield a 4-dimensional Kirkman cube, i.e. that there does not exist a set of 4 mutually orthogonal resolutions (of the considered type) with the property that any two form a $KS_3^*(39)$.

6. Conclusion.

The purpose of this article was to examine a special class of STSs of order 39 and to find those having at least two orthogonal resolutions. Although the class is reasonably restricted, it does admit a number of designs of the type we desired. It is hoped that the results of this paper can be generalized to give some new orders of $KS_3(v)$s and $SKC_3(v)$s and possibly an infinite class of direct constructions. At present the only orders less than 100 for which a $KS_3(v)$ is known to exist are $v = 27, 39, 63$ and 81. The only order less than 250 for which a $SKC_3(v)$ is known is $v = 39$.

Acknowledgements The authors would like to thank Mary Jayne Zettel for programming assistance. The first author would like to thank the Department of Combinatorics and Optimization, University of Waterloo, for its hospitality during the preparation of this paper. Research of the authors is supported by the Natural Sciences and Engineering Research Council of Canada under grants numbered A7268 and A9258.

References

[1] C.J. Colbourn, "Distinct cyclic Steiner triple systems", *Utilitas Mathematica* 22(1982), 103-126.

[2] J.H. Dinitz and D.R. Stinson, "The spectrum of Room cubes", *European Journal of Combinatorics* 2 (1981) 221-230.

[3] L. Heffter, "Über Tripelsysteme", *Mathematische Annalen* 49 (1897) 101-112.

[4] Z. Janko and T. Van Trung, "On projective planes of order 12 with an automorphism of order 13", *Geometriae Dedicata* 11 (1981), 257-284.

[5] R. Mathon and S.A. Vanstone, "On the existence of doubly resolvable Kirkman systems and equidistant permutation arrays", *Discrete Mathematics* 30 (1980) 157-172.

[6] R. Mathon, K. Phelps and A. Rosa, "Small Steiner triple systems and their properties", *Ars Combinatoria* 15 (1983) 3-110.

[7] R.C. Mullin and W.D. Wallis, "The existence of Room squares", *Aequationes Mathematicae* 13 (1975) 1-7.

[8] A. Rosa and S.A. Vanstone, "Kirkman cubes", *Annals of Discrete Mathematics* 18 (1983) 699-712.

[9] S.A. Vanstone and A. Rosa, "Starter-adder techniques for Kirkman squares and Kirkman cubes of small sides", *Ars Combinatoria* 14 (1982) 199-212.

Table 1.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11	6	1	9	13	4	10	15	7	12	8	18

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

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

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

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

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

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

This Page intentionally left blank

Annals of Discrete Mathematics 26 (1985) 321–334

Hill-Climbing Algorithms for the Construction of Combinatorial Designs

D. R. Stinson

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
CANADA

Abstract

In this paper we discuss the use of hill-climbing techniques in the construction of combinatorial designs.

1. Introduction

The construction of combinatorial designs has been and remains a very active area of research in discrete mathematics. It is often necessary to construct designs on the computer, and backtracking algorithms have been the traditional approach used. However, backtracking algorithms exhibit exponential behaviour, and become impractical for designs of even moderate size.

In certain instances, a hill-climbing approach can be used. This approach works very well in the case of Steiner triple systems. We describe a simple heuristic. A Steiner triple system is constructed block by block, and at no time in the algorithm is the number of blocks decreased. Although we cannot even guarantee that the algorithm will successfully construct a Steiner triple system, it appears to provide an extremely fast method of constructing these designs. Evidence suggests that a Steiner triple system with n points and $\frac{n(n-1)}{6}$ blocks can be constructed in time proportional to $n^2 \log n$.

We discuss the implementation of the above algorithm, and the numerical results we have obtained. Other similar problems are considered, including the construction of Latin squares, and strong starters. We also consider the completion of partial designs (an NP-complete problem), and the construction of non-isomorphic designs. Finally, we discuss when a hill-climbing approach is likely to succeed or fail. Many problems will not succumb to such an attack, but hill-climbing is nevertheless a technique which should be considered when using the computer to construct new combinatorial designs.

2. Hill-climbing and Steiner triple systems.

Let n be a positive integer. A *Steiner triple system* (or *STS*) of *order n* is a pair (X,B), where X is a set of n elements called *points*, and B is a set of 3-subsets of X (called *blocks*), such that every unordered pair of distinct points is contained in a unique block. It follows that there are $(n^2-n)/6$ blocks. A necessary condition for existence is that $n \equiv 1$ or 3 modulo 6, and indeed, this condition is also sufficient, as was demonstrated by Kirkman [5] in 1847.

There is a vast literature concerning the study of Steiner triple systems and their properties. (A comprehensive bibliography is given in [6]). Many constructions are known, both direct and recursive. In this section we investigate the generation of Steiner triple systems by computer. For purposes of comparison, we briefly describe a backtracking algorithm.

A *partial Steiner triple system* is a pair (X,B) , where as before, B is a set of blocks of size 3, but where every unordered pair of distinct blocks is contained in at most one block. The following is called a block-by-block backtracking algorithm:

```
Begin
b:=0;
B:=φ;
while b < (n²−n)/6  do
     if there is a B0 such that B∪{B0} is a partial STS then begin
          B:=B∪{B0};
          b:=b + 1 ;
          push (stack, B)
     end else begin
          b:=b − 1 ;
          B:=B \ pop (stack)
     end
end.
```

The time required for such a backtracking algorithm to successfully construct an *STS* is an exponential function of n. This remains true even if refinements such as look-ahead are included. The basic reason for this is that too much time is wasted investigating "dead-ends".

A hill-climbing approach solves this difficulty. An algorithm could work as follows:

```
Begin
b:=0;
B:=φ;
while b < (n²−n)/6  do
     if there is a B0 such that  B∪{B0}  is
```

```
a partial STS then begin
     B:=B∪{B_0} ;
     b:=b + 1
end else begin
     find B_1 ∈ B and B_0 ∉ B such that
     B∪{B_0}\{B_1} is a partial STS ;
     B:=B∪{B_0}\{B_1}
end
end.
```

The most important feature of the above algorithm is that the backtracking step has been eliminated; the size of a partial *STS* is never decreased in the course of the algorithm.

We must discuss what happens when we cannot add a block to a partial *STS*. We alter B slightly, by replacing a block B_1 with another block B_0. This can be done as follows. First choose a point x which has not yet occurred with all other points (such a point is called a *live* point). There must be two points y and z with which x has not occurred. We let $B_0 = \{x,y,z\}$. If y and z have not yet occurred together, then we could have added B_0 as a new block. Hence they have occurred in a block, which we name B_1. Then we perform the *switching operation*, replacing B_1 by B_0.

Since this switching operation is so easy to perform, it is not worth our while to check that there is no way to extend the partial *STS* before doing the switching operation. So our algorithm works as follows:

```
Begin
b:=0;
B:=φ
while b < (n^2−n)/6 do begin
     choose a live point x;
     choose y, z which have not occurred with x;
     B_0:={x,y,z}
     if y, z have not occurred in a block of B then begin
          B:=B∪{B_0};
          b:= b + 1
     end else begin
          B_1:= the block of B which contains y, z ;
          B:= B∪{B_0}\{B_1}
     end
end
end.
```

An *iteration* consists of either extending the partial *STS* or performing a switching operation. If we are careful, we can implement this algorithm so that the time taken per iteration is constant (i.e. not an increasing function of either n or b).

We need to keep a table of all live points. This table will not be ordered, so we require an array which indicates where in the table a given point occurs (this is necessary for updating operations). When a point ceases to live, the last point in the table is moved to occupy its place. If a point which is not live becomes live again, it is simply added to the end of the table. Hence, the operation "choose a live point" consists just of generating a random integer between 1 and the number of live points, and choosing the point in the given position of the table.

For each live point we need a table of points which have not occurred with that point, and an array indicating where in the table a given point occurs. These are maintained in a fashion similar to the table of live points.

Of course, we have to keep track of the "current" partial *STS* we construct. We also need to know the block which contains any given pair, in order to perform a switching operation.

Thus the total memory required is proportional to n^2, and the total time required is proportional to the number of iterations. Unfortunately, we are unable to prove any theoretical results concerning this number. It is even conceivable that the algorithm will sometimes not terminate. However, this does not seem to occur in practice. Using assumptions that blocks are independent of each other (which is clearly not true), one would suspect that the number of iterations is proportional to $n^2 \log n$. This appears to be a good estimate. Our results are recorded in Table 1. (We use b to denote $(n^2-n)/6$). The algorithm was programmed using Pascal/VS and run on the University of Manitoba Amdahl 470-V7 computer. Ten *STS* of each order were constructed.

3. Related Problems.

In this section we discuss several other combinatorial design problems to which hill-climbing can be applied.

A *Latin square* of *order* n is an n by n array of the integers $1,\ldots,n$, in which each integer occurs once in each row and each column. If we label the rows r_i $(1 \le i \le n)$, and we label the columns c_i $(1 \le i \le n)$, then we can write down n^2 triples, each of the form $\{r_i, c_j, k\}$. Each such triple forms a transversal of the three sets $R = \{r_i\}$, $C = \{c_i\}$, and $S = \{i\}$, and given two elements from different sets, there is a unique triple containing them. Such a collection of triples is called a transversal design, and a Latin square can be constructed from any transversal design by letting the three sets represent (in

Table 1					
Construction of Steiner Triple Systems					
n	*avg. # of iterations*	$b = \frac{(n^2-n)}{6}$	*avg. time (sec.)*	$\frac{\textit{avg. msec.}}{\textit{iteration}}$	$\frac{\textit{avg.\# iterations}}{(b \log b)}$
31	155	486	.157	.323	.621
61	610	2317	.437	.188	.592
91	1365	5588	1.52	.272	.567
121	2420	9753	2.64	.271	.517
151	3775	15830	2.85	.180	.509
181	5430	23064	4.12	.178	.493
211	7385	32129	5.79	.180	.488
241	9640	41430	6.35	.153	.468
271	12195	54267	12.04	.221	.472

any order) rows, columns, and symbols, and filling one cell of the Latin square for each triple of the transversal design.

One can construct a transversal design by a hill-climbing method, using a heuristic very similar to that used for *STS*. If R, C, and S represent the three sets, we can easily find r_i, c_j, and k, so that at most one pair has occurred in a given partial transversal design. If a pair has already occurred, then perform the switching operation, as before.

A more difficult problem is the construction of strong starters. Let $n = 2t + 1$ be an odd positive integer. A *strong starter* in $\mathbb{Z}_n$ is a set $S = \{ \{x_i, y_i\} : 1 \le i \le t \}$ which satisfies

(i) $\{x_i, y_i : 1 \le i \le t\} = \mathbb{Z}_n \setminus \{0\}$,

(ii) $\{\pm(x_i - y_i) : 1 \le i \le t\} = \mathbb{Z}_n \setminus \{0\}$,

(iii) $x_i + y_i \ne x_j + y_j$ if $i \ne j$, and $x_i + y_i \ne 0$, for any i .

Strong starters are used extensively for the construction of Room squares, Howell designs, one-factorizations of complete graphs, and related objects. It is suspected, but still unproven, that there exists a strong starter of any odd order $n \ge 11$.

Backtracking algorithms break down by order 100, becoming impractical (see [2]). In [3], Dinitz and Stinson describe a hill-climbing algorithm for the construction of strong starters. Here one heuristic does not appear to be sufficient. However, several heuristics are described, and incorporated into an algorithm which uses all of them. The algorithm does not succeed approximately 10% of the time. However, when it does succeed, it appears to require approximately $n \log n$ iterations (applications of the heuristics). In [3], the implementation was not as efficient as possible (there were some linear searches, which are very inefficient). However, this algorithm can be programmed so that each iteration takes constant time. With a more efficient programme, the results in Table 2 were obtained (two strong starters of each order were constructed).

Table 2			
Construction of Strong Starters			
n	average # iterations	average time(sec)	time/iteration
1001	8570	.65	$.75 \times 10^{-4}$
3001	21620	1.31	$.60 \times 10^{-4}$
5001	28624	1.75	$.61 \times 10^{-4}$
8001	56550	3.67	$.65 \times 10^{-4}$
10001	95524	7.05	$.73 \times 10^{-4}$

These times are a significant improvement over those obtained in [3], where, for example, it took 58 seconds to construct a strong starter of order 10001.

Another interesting question is the completion of partial designs: given a partial *STS* (X,B), is there an *STS* (X,B_1) such that $B \subseteq B_1$? This problem is NP-complete [1]. (Also, the problem of completing a partial Latin square is NP-complete). We can try to complete a partial design using the same heuristic as before, except that some switching operations are not allowed - the blocks of the partial design cannot be altered. We suspect that, if a partial design can be completed, this method will either find a completion quite quickly, or reach a "dead end" from which it cannot escape. Repeated applications of the algorithm should, in most cases, provide a completion of any design which can be completed.

We have tried to complete partial Steiner triple systems by this method, with differing amounts of success (one can certainly do far better by this approach than by backtracking).

First, we generate a partial *STS* containing a certain number of blocks, which we denote by FIXED. We then attempt to complete this partial design. We specify a maximum number of iterations (which depends on r and FIXED) denoted by NITER. If the design is not completed in NITER iterations, we quit and start over. If a given partial design is not completed in 10 tries, we abandon it. We are thus allowing for the possibility of "dead ends" caused by the existence of the fixed blocks.

We observe a very interesting phenomenon. The probability of successfully completing a partial design by this method (as a function of *FIXED*) is at first very close to 1, and a certain later point, drops very rapidly to 0. For $n=43$ $(b=301)$ we find the results given in Table 3.

Table 3		
Completion of partial designs		
FIXED	*Percentage of blocks FIXED*	*Probability of successful completion* 10 *tries* for *each design*
150	50	.98
155	51.6	.83
160	53.3	.50
165	55	0.0

What the above results do not show is why the uncompleted designs were not complete. Some of them may in fact be completable, even though the algorithm was unsuccessful. To test this possibility, we did the following. An *STS* is generated, and then a random subset of blocks is selected to be our partial designs. Such a partial design is completable, so we hope our algorithm will succeed. We find that the probability of completing such a partial design in any given try (as a function of FIXED) is at first very close to 1, then drops to a minimum (of approximately .005) and then later increases very quickly back to 1. For $n=43$ $(b=301)$, our results are tabulated below. (For each value of FIXED, at least 30 designs were considered).

Intuitively, these results seem reasonable. When FIXED is small, there is no difficulty completing the partial design. As FIXED is increased, there are fewer switching operations possible, and it is more likely that we reach a "dead end". As FIXED is increased further, there are still fewer possible switching operations. But there is at least one completion, so the correct switching operations are "forced".

Even in the most difficult cases (where FIXED is between 190 and 200), repeated application of this approach would eventually yield a completion. Particular examples have required over 100 tries before a completion was found.

The last problem we consider is the generation of non-isomorphic *STS*. Two *STS* $(X_1 B_1)$ and (X_2,B_2) are said to be *isomorphic* if there is a bijection $\phi : X_1 \to X_2$ such that $\{x,y,z\} \in B_1$ if and only if $\{\phi(x), \phi(y), \phi(z)\} \in B_2$.

Table 4		
Completion of partial designs, all completable		
FIXED	*Percentage of blocks FIXED*	*Probability of successful completion*
125	41.5	.923
130	43.2	.850
135	44.8	.729
140	46.5	.667
145	48.1	.526
150	49.8	.441
155	51.5	.354
160	53.2	.300
165	54.8	.100
170	56.5	.069
180	59.8	.037
190	63.1	.005
200	66.4	.005
210	69.7	.054
215	71.4	.368
220	73.1	.500
225	74.7	.684
230	76.4	.768
240	79.7	.969

There is an algorithm to test isomorphism of *STS* in subexponential time, but there is no known polynomial algorithm. In practice, one often proves that two designs are non-isomorphic by the use of invariants, many of which can be found in polynomial time.

One invariant, called a *fragment vector,* is discussed in [4]. A *fragment* in an *STS* is a set of four blocks, and six points, in which any two blocks contain a common point, and any point occurs in two of the four blocks. For each point x, let $f(x)$ denote the number of fragments containing x. The *fragment vector* is a list of the integers $f(x)$ in non-decreasing order. Clearly, isomorphic *STS* have the same fragment vectors. Also, one can enumerate all fragments in an *STS* of order n in time proportional to n^3, so it is a fairly fast invariant. For triple systems up to order 15, it is also effective: two *STS* of order $n \leq 15$ are isomorphic if and only if they have the same fragment vectors.

We have investigated *STS* of order 15, generated by our algorithm, by means of fragment vectors. One would hope that an *STS* generated by a hill-climbing technique is a random *STS*. If we make a list of all *STS* of order 15 (on a fixed symbol set) they can be collected into 80 isomorphism classes $C_1,...,C_{80}$ (see [6]). The size of class C_i is $15!/|G_i|$, where G_i is the group of automorphisms of any design in class C_i . A truly random algorithm would produce an *STS* in class C_i with probability

$$\frac{1}{|G_i| \sum_{j=1}^{80} \frac{1}{|G_j|}}.$$

We generated and classified 10000 *STS* of order 15. The total time taken was 143 seconds, so designs are constructed and classified at the rate of over 70 per second. Our results are presented in Table 5. (The numbering of the *STS* is the "traditional" numbering, as is followed in [6]). The "expected" values are calculated according to the above probabilities. We do not obtain an acceptable goodness of fit. Designs with large automorphism groups are not constructed as often as we would expect. Nevertheless, there is a good overall correlation between the observed values and the reciprocal of the order of the automorphism group.

We feel that hill-climbing, used in conjunction with fragment vectors, provides a very good method of generating large numbers of non-isomorphic *STS* of a given order. One can retain in memory a binary tree of fragment vectors (ordered lexicographically). When an *STS* is generated, it can be checked very quickly whether it has a new fragment vector. If so, then the *STS* can be written onto a tape or disk for future use. The use of an invariant provides a significant saving in both time and memory. Of course, the invariant will fail to distinguish between certain non-isomorphic *STS*.

Table 5			
10000 Steiner triple systems of order 15			
System Number	*Order of automorphism group*	*Number of designs expected*	*Number of designs observed*
1	20160	0	0
2	192	1	0
3	96	2	0
4	8	27	5
5	32	7	2
6	24	9	5
7	288	1	1
8	4	54	25
9	2	108	68
10	2	108	68
11	2	108	80
12	3	72	30
13	8	27	13
14	12	18	10
15	4	54	37
16	168	1	0
17	24	9	10
18	4	54	33
19	12	18	12
20	3	72	52
21	3	72	43
22	3	72	64
23	1	217	199
24	1	217	190
25	1	217	176
26	1	217	164
27	1	217	214

Table 5 (continued)			
10000 Steiner triple systems of order 15			
System Number	*Order of automorphism group*	*Number of designs expected*	*Number of designs observed*
28	1	217	189
29	3	72	63
30	2	108	100
31	4	54	50
32	1	217	200
33	1	217	221
34	1	217	190
35	3	72	69
36	4	54	56
37	12	18	23
38	1	217	202
39	1	217	206
40	1	217	179
41	1	217	217
42	2	108	130
43	6	36	36
44	2	108	116
45	1	217	249
46	1	217	252
47	1	217	228
48	1	217	259
49	1	217	237
50	1	217	243
51	1	217	209
52	1	217	238
53	1	217	249
54	1	217	209

Table 5 (continued)			
10000 Steiner triple systems of order 15			
System Number	*Order of automorphism group*	*Number of designs expected*	*Number of designs observed*
55	1	217	243
56	1	217	233
57	1	217	254
58	1	217	254
59	3	72	48
60	1	217	255
61	21	10	12
62	3	72	90
63	3	72	86
64	3	72	67
65	1	217	236
66	1	217	230
67	1	217	217
68	1	217	258
69	1	217	267
70	1	217	236
71	1	217	239
72	1	217	231
73	4	54	58
74	4	54	71
75	3	72	91
76	5	43	45
77	3	72	85
78	4	54	64
79	36	6	3
80	60	4	6

We have generated *STS* of order 19 in this fashion. 4000 *STS* were generated and classified according to fragment vectors. (The time taken was 108 seconds, a rate of about 37 per second). 3645 distinct fragment vectors were found, so we know that over 90% of the *STS* generated are non-isomorphic. There are, in fact, over 280000 non-isomorphic *STS* of order 19, so we expect that most of the remaining 355 *STS* are also non-isomorphic. The number 3645 is dependent on two factors: the tendency of the hill-climbing algorithm to generate non-isomorphic designs, and the effectiveness of the invariant. It is interesting to note that, of the first 500 *STS* generated, 496 had distinct fragment vectors.

4. Discussion.

There is a metatheorem among combinatorialists that, for any given class of designs, there is an integer N such that one can solve the case N by hand, the case $N+1$ by computer, and the case $N+2$ cannot be done. This is more formally referred to as "the combinatorial explosion" and indicates the futility of back-tracking methods for constructing designs.

Hill-climbing exemplifies a completely different philosophy from back-tracking. Hill-climbing is non-enumerative whereas backtracking (in theory) finds all solutions. Hill-climbing implicitly assumes the existence of a solution, whereas backtracking can (in theory) prove that no solution exists. These points give some clue as to when hill-climbing is a feasible technique: there must be a solution, and, most likely, there must be many solutions.

However, the overriding factor is the heuristic or heuristics used in the algorithm to "build up" the design. The heuristics should be fast and applicable in any situation. In the situations where hill-climbing has not proved effective (see [8] and [9]), the problem is the difficulty of finding good heuristics.

In the problems investigated in this paper, we had a very simple, fast, effective heuristic. For more difficult design problems, perhaps a combination of hill-climbing and back-tracking can be used.

Hill-climbing is a technique which has been useful in many types of optimization problems (see [7]); it is our hope that it will prove useful in the study of combinatorial designs.

References

[1] C. J. Colbourn, "Embedding partial Steiner triple systems is NP-complete", *Journal of Combinatorial Theory* A35 (1983) 100-105.

[2] R. J. Collens and R. C. Mullin, "Some properties of Room squares - a computer search", Proceedings of the First Louisiana Conference on Combinatorics, Graph Theory, and Computing, Baton Rouge, 1970, 87-111.

[3] J. H. Dinitz and D. R. Stinson, "A fast algorithm for finding strong starters", *SIAM Journal of Algebraic and Discrete Methods* 2 (1981) 50-56.

[4] P. B. Gibbons, "Computing techniques for the construction and analysis of block designs", Ph.D. Thesis, University of Toronto, 1976.

[5] T. P. Kirkman, "On a problem in combinations", *Cambridge and Dublin Mathematical Journal* (1847) 191-204.

[6] R. A. Mathon, K. T. Phelps and A. Rosa, "Small Steiner systems and their properties", *Ars Combinatoria* 15 (1983) 3-110.

[7] C. Papadimitriou and K. Steiglitz, Chapter 19 in *Combinatorial Optimization: Algorithms and Complexity* Prentice Hall, Englewood Cliffs, N.J., 1982.

[8] D. P. Shaver, "Construction of (v,k,λ)-configurations using a non-enumerative search technique", Ph.D. Thesis, Syracuse University, 1973.

[9] M. Tompa, "Hill-climbing: a feasible search technique for the construction of combinatorial configurations", M.Sc. Thesis, University of Toronto, 1975.

ANNALS OF DISCRETE MATHEMATICS

Vol. 1: Studies in Integer Programming
edited by P.L. HAMMER, E.L. JOHNSON, B.H. KORTE and G.L. NEMHAUSER
1977 viii + 562 pages

Vol. 2: Algorithmic Aspects of Combinatorics
edited by B. ALSPACH, P. HELL and D.J. MILLER
1978 out of print

Vol. 3: Advances in Graph Theory
edited by B. BOLLOBÁS
1978 viii + 296 pages

Vol. 4: Discrete Optimization, Part I
edited by P.L. HAMMER, E.L. JOHNSON and B. KORTE
1979 xii + 300 pages

Vol. 5: Discrete Optimization, Part II
edited by P.L. HAMMER, E.L. JOHNSON and B. KORTE
1979 vi + 454 pages

Vol. 6: Combinatorial Mathematics, Optimal Designs and their Applications
edited by J. SRIVASTAVA
1980 viii + 392 pages

Vol. 7: Topics on Steiner Systems
edited by C.C. LINDNER and A. ROSA
1980 x + 350 pages

Vol. 8: Combinatorics 79, Part I
edited by M. DEZA and I.G. ROSENBERG
1980 xxii + 310 pages

Vol. 9: Combinatorics 79, Part II
edited by M. DEZA and I.G. ROSENBERG
1980 viii + 310 pages

Vol. 20: Convexity and Graph Theory
edited by M. ROSENFELD and J. ZAKS
1984 xii + 340 pages

Vol. 21: Topics on Perfect Graphs
edited by C. BERGE and V. CHVÁTAL
1984 xiv + 370 pages

Vol. 22: Trees and Hills: Methodology for Maximizing Functions of Systems of Linear Relations
R. GREER
1984 xiv + 352 pages

Vol. 23: Orders: Description and Roles
edited by M. POUZET and D. RICHARD
1984 xxviii + 548 pages

Vol. 24: Topics in the Theory of Computation
edited by M. KARPINSKI and J. VAN LEEUWEN
1984 x + 188 pages

Vol. 25: Analysis and Design of Algorithms for Combinatorial Problems
edited by G. AUSIELLO and M. LUCERTINI
1985 x + 320 pages

This Page intentionally left blank