Contents

Chapter 1
Designs, groups and computing

Leonard H. Soicher

1.1 Introduction

In this chapter we present some applications of groups and computing to the discovery, construction, classification and analysis of combinatorial designs. The focus is on certain block designs and their statistical efficiency measures, and in particular semi-Latin squares, which are certain 1-designs with additional block structure and which generalise Latin squares.

The chapter starts with background material on block designs, statistical efficiency measures for 1-designs, permutation groups, Latin squares and semi-Latin squares. We then review statistical optimality results for semi-Latin squares. Next, we introduce the recent theory of "uniform" semi-Latin squares, which generalise complete sets of mutually orthogonal Latin squares, and are statistically "Schur-optimal". We then describe a recent construction which determines a semi-Latin square $\mathrm{SLS}(G)$ from a transitive permutation group G, the square being uniform precisely when G is 2-transitive. Moreover, we show how certain structural properties of $\mathrm{SLS}(G)$ are determined from the structure of G, and (due to Martin Liebeck), how statistical efficiency measures of $\mathrm{SLS}(G)$ are determined from the degrees and multiplicities of the irreducible constituents of the permutation character of G.

We then turn to computation, and discuss the DESIGN package [34] for GAP [18], focussing on the function `BlockDesigns`, which can be used for a wide variety of block design classifications using groups, and the function `BlockDesign-Efficiency`, for exact information on the efficiency measures of a given 1-design. We then show how the `BlockDesigns` function can be applied to obtain classifications of semi-Latin squares, and of uniform semi-Latin squares and their subsquares, which can then be analysed using the `BlockDesignEfficiency` function. We give an extended example of this, including the determination of the first published efficient $(6\times 6)/k$ semi-Latin squares, for $k = 7, 8, 9$. It is hoped that our examples of the use of the DESIGN package will help the reader to use this package in their own investigations of designs. We conclude the chapter with some open problems.

1.2 Background material

1.2.1 Block designs

A *block design* is an ordered pair $(V,\mathscr{B})$, such that V is a finite non-empty set of *points*, and $\mathscr{B}$ is a (disjoint from V) finite multiset (or collection) of non-empty subsets of V called *blocks*, such that every point is in at least one block.

In a multiset (of blocks say), order does not matter, but the number of times an element occurs (its *multiplicity*) does indeed matter. We denote a multiset with elements $A_1,\dots,A_b$ (including any repeated elements) by $[A_1,\dots,A_b]$. For example, the block design $(V,\mathscr{B})$, with $V=\{1,2,3\}$ and

$$\mathscr{B}=[\{1,2\},\{1,2,3\},\{1,2,3\},\{1,3\},\{2,3\}],$$

has three points and five blocks.

Let t be a non-negative integer. A *t-design*, or more specifically a t-(v,k,λ) *design*, is a block design $(V,\mathscr{B})$ such that $v=|V|$, each block has the same size k, and each t-subset of V is contained in the same positive number λ of blocks. It is well-known that a t-design is also an s-design, for $s=0,\dots,t-1$ (see, for example, [23, Theorem 19.3]). For example, the block design $(V,\mathscr{B})$, with $V=\{1,\dots,7\}$ and

$$\mathscr{B}=[\{1,2,3\},\{1,4,5\},\{1,6,7\},\{2,4,7\},\{2,5,6\},\{3,4,6\},\{3,5,7\}],$$

is a 2-$(7,3,1)$ design, and also a 1-$(7,3,3)$ design and a 0-$(7,3,7)$ design.

The dual of a block design is obtained by interchanging the roles of points and blocks. More precisely, if $\Delta=(\{\alpha_1,\dots,\alpha_v\},[A_1,\dots,A_b])$ is a block design, then the *dual* Δ^* of Δ is the block design $(\{1,\dots,b\},[B_1,\dots,B_v])$, with $B_i=\{j:\alpha_i\in A_j\}$. Note that if Δ is a 1-(v,k,r) design then Δ^* is a 1-$(vr/k,r,k)$ design.

Two block designs $\Delta_1=(V_1,\mathscr{B}_1)$ and $\Delta_2=(V_2,\mathscr{B}_2)$ are *isomorphic* if there is a bijection from V_1 to V_2 that maps $\mathscr{B}_1$ to $\mathscr{B}_2$. The set of all isomorphisms from a block design Δ to itself forms a group, the *automorphism group* $\mathrm{Aut}(\Delta)$ of Δ.

Block designs are of interest to both pure mathematicians and statisticians. They are used by statisticians for the design of comparative experiments: the points represent “treatments” to be compared, and the blocks represent homogeneous testing material, so usually, only inter-block information is used in the comparison of treatment effects. Although statisticians sometimes allow points to be repeated within a block, here we do not.

1.2.2 *Efficiency measures of* 1*-designs*

We now discuss certain statistical efficiency measures of 1-designs. The basic idea is that once you have decided that your experimental design needs to be in a certain class $\mathscr{C}$ of 1-(v,k,r) designs, you want to choose a design in $\mathscr{C}$ able to give as much information as possible; that is, the most "efficient" design in $\mathscr{C}$ with respect to one or more of the efficiency measures defined below. The reader who wants to learn more about statistical design theory and the theory of optimal designs should consult the excellent survey article [6], which was written for combinatorialists. Other useful references for these topics include [3, 4, 11, 29].

Let Δ be a 1-(v,k,r) design, with $v \geq 2$. The *concurrence matrix* of Δ is the $v \times v$ matrix whose rows and columns are indexed by the points, and whose (α, β)-entry is the number of blocks containing both points α and β. The *scaled information matrix* of Δ is

$$F(\Delta) := I_v - (rk)^{-1}\Lambda,$$

where I_v is the $v \times v$ identity matrix and Λ is the concurrence matrix of Δ. The matrix $F(\Delta)$ is real, symmetric and positive semi-definite, and is scaled so that its (all real) eigenvalues lie in the interval $[0,1]$. Moreover, $F(\Delta)$ has constant row-sum 0, so the all-1 vector is an eigenvector with corresponding eigenvalue 0. It can be shown that the remaining eigenvalues are all non-zero if and only if Δ is connected (i.e. its point-block incidence graph is connected). Omitting the zero eigenvalue corresponding to the all-1 vector, the eigenvalues

$$\delta_1 \leq \delta_2 \leq \cdots \leq \delta_{v-1}$$

of $F(\Delta)$ are called the *canonical efficiency factors* of Δ. Note that the canonical efficiency factors of two isomorphic 1-designs are the same.

If Δ is not connected, then we define $A_\Delta = D_\Delta = E_\Delta := 0$. Otherwise, we define these *efficiency measures* by

$$\begin{aligned} A_\Delta &:= (v-1)/\sum_{i=1}^{v-1} 1/\delta_i, \\ D_\Delta &:= \left(\prod_{i=1}^{v-1}\delta_i\right)^{1/(v-1)}, \\ E_\Delta &:= \delta_1 = \min\{\delta_1, \ldots, \delta_{v-1}\}. \end{aligned}$$

Note that A_Δ (respectively D_Δ) is the harmonic mean (respectively geometric mean) of the canonical efficiency factors of Δ.

If $k = v$ (so each block contains every point) then each canonical efficiency factor of Δ is equal to 1 and so are each of the efficiency measures above. If $k < v$, we want to minimize the loss of "information" due to being forced to use "incomplete" blocks, and want the above efficiency measures to be as close to 1 as possible.

We now define optimality in a class of 1-designs with respect to a given efficiency measure. The 1-(v,k,r) design Δ is *A-optimal* in a class $\mathscr{C}$ of 1-(v,k,r) designs containing Δ if $A_\Delta \geq A_\Gamma$ for each $\Gamma \in \mathscr{C}$. D-optimal and E-optimal are defined similarly. We say Δ is *Schur-optimal* in a class $\mathscr{C}$ of 1-(v,k,r) designs containing Δ if for each design $\Gamma \in \mathscr{C}$, with canonical efficiency factors $\gamma_1 \leq \cdots \leq \gamma_{v-1}$, we have

$$\sum_{i=1}^{\ell} \delta_i \geq \sum_{i=1}^{\ell} \gamma_i,$$

for $\ell = 1,\ldots,v-1$. A Schur-optimal design need not exist within a given class $\mathscr{C}$ of 1-(v,k,r) designs, but when it does, that design is optimal in $\mathscr{C}$ with respect to a very wide range of statistical optimality criteria, including being A-, D- and E-optimal [19].

It is not difficult to see that if Δ is a 2-(v,k,λ) design, then the canonical efficiency factors of Δ are all equal (to $v(k-1)/((v-1)k)$), from which it follows that Δ is Schur-optimal in the class of all 1-$(v,k,\lambda(v-1)/(k-1))$ designs. However, a 2-design may well not exist with the properties we are interested in.

The canonical efficiency factors of a 1-design Δ not equal to 1, and their multiplicities, are the same as those of the dual block design Δ^* of Δ (see [6, Section 3.1.1]). We thus obtain the following:

Theorem 1.1. *A* 1-(v,k,r) *design* Δ *is A-optimal (respectively D-optimal, E-optimal, Schur optimal) in a class* $\mathscr{C}$ *of* 1-(v,k,r) *designs if and only if* Δ^* *is A-optimal (respectively D-optimal, E-optimal, Schur optimal) in the class consisting of the dual block designs of the designs in* $\mathscr{C}$.

1.2.3 Permutation groups

We now review some basic definitions and results in the theory of permutation groups and group actions. An excellent reference for permutation groups is [10].

A *permutation group* G on a set V of *points* is a subgroup of the group Sym(V) of all permutations of V, with the group operation being composition. The image of $\alpha \in V$ under $g \in G$ is denoted αg (our permutations act on the right). The *degree* of G is the size of V. The *symmetric group of degree n*, denoted S_n, is the group Sym($\{1,\ldots,n\}$) of all permutations of $\{1,\ldots,n\}$.

An *action* of a group G on a set V is a function $\psi : V \times G \to V$, with $(\alpha,g)\psi$ denoted α^g, such that $\alpha^1 = \alpha$ and $(\alpha^g)^h = \alpha^{gh}$, for all $\alpha \in V$ and all $g,h \in G$. Given an action of G on V, we say that G *acts on* V.

If G is a permutation group on V then G acts *naturally* on V, where $\alpha^g := \alpha g$. An action of G on V gives rise to a homomorphism $\phi : G \to \mathrm{Sym}(V)$ defined by $\alpha(g\phi) := \alpha^g$ for all $\alpha \in V$ and $g \in G$.

Suppose G acts on V and $\alpha \in V$. The *G-orbit* of α is $\alpha^G := \{\alpha^g : g \in G\}$. The set $\{\alpha^G : \alpha \in V\}$ of all G-orbits is a partition of V. The *stabilizer* in G of α is

$G_\alpha := \{g \in G : \alpha^g = \alpha\}$. This stabilizer G_α is a subgroup of G, and if G is finite then $|G| = |G_\alpha||\alpha^G|$.

Let G act on a finite set V. Then G has actions on many sets. Here are some:

- G acts on the set of all t-tuples of distinct elements of V, where $(\alpha_1,\ldots,\alpha_t)^g := (\alpha_1^g,\ldots,\alpha_t^g)$;
- G acts on the set of all subsets of V, where $\{\alpha_1,\ldots,\alpha_k\}^g := \{\alpha_1^g,\ldots,\alpha_k^g\}$;
- G acts on the set of all finite multisets of subsets of V, where $[A_1,\ldots,A_b]^g := [(A_1)^g,\ldots,(A_b)^g]$;
- G acts on the set of all block designs with point set V, where $(V,\mathscr{B})^g := (V,\mathscr{B}^g)$.

A permutation group G on a non-empty set V is *transitive* if for every $\alpha,\beta \in V$ there is a $g \in G$ with $\alpha g = \beta$ (i.e. there is just one G-orbit in the natural action of G on V). More generally, a permutation group G on a set V of size at least t is *t-transitive* if for every pair $(\alpha_1,\ldots,\alpha_t)$, $(\beta_1,\ldots,\beta_t)$ of t-tuples of distinct elements of V, there is a $g \in G$ with $(\alpha_1 g,\ldots,\alpha_t g) = (\beta_1,\ldots,\beta_t)$. It is easy to see that if G is a t-transitive permutation group on a finite set V of size $v \geq t$, and B is a subset of V of size $k \geq t$, then the block design with point set V and block (multi)set the G-orbit B^G is a t-(v,k,λ) design, with $\lambda = |B^G|\binom{k}{t}/\binom{v}{t}$.

1.2.4 Latin squares

A *Latin square* of *order* n is an $n \times n$ array L, whose entries are elements of an n-set Ω, the set of *symbols* for L, such that each symbol occurs exactly once in each row and exactly once in each column of L. For example, a completed Sudoku puzzle is a special kind of Latin square of order 9.

Two Latin squares L_1 and L_2 of order n, with respective symbol-sets Ω_1 and Ω_2, are *orthogonal* if for every $\alpha_1 \in \Omega_1$ and $\alpha_2 \in \Omega_2$, there is an (i,j) such that α_1 is the (i,j)-entry in L_1 and α_2 is the (i,j)-entry in L_2. For example, here are two orthogonal Latin squares of order 3:

1	2	3
3	1	2
2	3	1

4	5	6
5	6	4
6	4	5

Latin squares $L_1,\ldots,L_m$ of order n are said to be *mutually orthogonal* if they are pairwise orthogonal, in which case $\{L_1,\ldots,L_m\}$ is called a set of *mutually orthogonal Latin squares* or a set of *MOLS*.

Let $n > 1$. A set of MOLS of order n has size at most $n-1$, and the existence of a set of MOLS of order n having size $n-1$ (called a *complete set* of MOLS) is equivalent to the existence of a projective plane of order n. A complete set of MOLS of order n exists when n is a prime power, but it is a famous problem as to whether a complete set of MOLS of order n exists for some non-prime-power n.

1.2.5 Semi-Latin squares

An $(n\times n)/k$ *semi-Latin square* is an $n\times n$ array S, whose entries are k-subsets of an nk-set Ω, the set of *symbols* for S, such that each symbol is in exactly one entry in each row and exactly one entry in each column of S. The entry in row i and column j is called the (i,j)-entry of S and is denoted by $S(i,j)$. To avoid trivialities, we assume throughout that $n>1, k>0$. Note that an $(n\times n)/1$ semi-Latin square is (essentially) the same thing as a Latin square of order n. We consider two $(n\times n)/k$ semi-Latin squares to be *isomorphic* if one can be obtained from the other by applying one or more of: a row permutation, a column permutation, transposing, and renaming symbols.

Semi-Latin squares have many applications, including the design of agricultural experiments, consumer testing, and message authentication (see [2, 5, 17, 26]). Semi-Latin squares exist in profusion, and a good choice of semi-Latin square for a given application can be very important.

Let s be a positive integer. An *s-fold inflation* of an $(n\times n)/k$ semi-Latin square is obtained by replacing each symbol α in the semi-Latin square by s symbols $\sigma_{\alpha,1},\ldots,\sigma_{\alpha,s}$, such that $\sigma_{\alpha,i}=\sigma_{\beta,j}$ if and only if $\alpha=\beta$ and $i=j$. The result is an $(n\times n)/(ks)$ semi-Latin square. For example, here is a $(3\times 3)/2$ semi-Latin square formed by a 2-fold inflation of a Latin square of order 3:

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

The *superposition* of an $(n\times n)/k$ semi-Latin square with an $(n\times n)/\ell$ semi-Latin square (with disjoint symbol sets) is performed by superimposing the first square upon the second, resulting in an $(n\times n)/(k+\ell)$ semi-Latin square. For example, here is a $(3\times 3)/2$ semi-Latin square which is the superposition of two (mutually orthogonal) Latin squares of order 3:

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

(1.1)

An $(n\times n)/k$ semi-Latin square in which any two distinct symbols occur together in at most one block is called a SOMA(k,n) (SOMA is an acronym for "simple orthogonal multi-array" [25]). For example, the semi-Latin square (1.1) is a SOMA$(2,3)$, and more generally, the superposition of k MOLS of order n is a SOMA(k,n). However, a SOMA(k,n) need not be a superposition of MOLS, and may exist even when there do not exist k MOLS of order n. However, it is easy to see that if a SOMA(k,n) exists then $k<n$.

The *dual* S^* of an $(n\times n)/k$ semi-Latin square S is a block design with point set $\{1,\ldots,n\}^2$ and nk blocks, one for each symbol of S, with the block for a symbol α consisting precisely of the ordered pairs (i,j) such that $\alpha\in S(i,j)$. Each block of

S^* is of the form $[(1,1^g),\dots,(n,n^g)]$ for some $g \in S_n$, and S^* is a 1-(n^2,n,k) design, which may have repeated blocks. Up to the naming of its symbols, the semi-Latin square S can be recovered from S^*, so S^* really represents the class of semi-Latin squares obtainable from S by renaming symbols. For example, let S be the semi-Latin square (1.1). Then $S^* = (V,\mathscr{B})$, with $V = \{1,2,3\}^2$ and

$$\mathscr{B} = \begin{array}{l} [\{(1,1),(2,2),(3,3)\},\ \{(1,2),(2,3),(3,1)\}, \\ \ \{(1,3),(2,1),(3,2)\},\ \{(1,1),(2,3),(3,2)\}, \\ \ \{(1,2),(2,1),(3,3)\},\ \{(1,3),(2,2),(3,1)\}]. \end{array}$$

1.3 Optimality results for semi-Latin squares

Let S be an $(n \times n)/k$ semi-Latin square. If we ignore the row and column structure of S, we obtain its *underlying block design* $\Delta(S)$, whose points are the symbols of S and whose blocks are the entries of S. Note that $\Delta(S)$ is a 1-(nk,k,n) design, and that the dual S^* of S is isomorphic as a block design to the dual of $\Delta(S)$. However, S^* has a structured point-set, unlike $\Delta(S)^*$.

Following the analysis of Bailey [2], S is *optimal* with respect to a given statistical optimality criterion if and only if $\Delta(S)$ is optimal with respect to that criterion in the class of underlying block designs of $(n \times n)/k$ semi-Latin squares. For this reason, we say that S has canonical efficiency factors, or a given efficiency measure, when $\Delta(S)$ has those canonical efficiency factors, or that efficiency measure.

Various optimality results for $(n \times n)/k$ semi-Latin squares are known. These include:

- Cheng and Bailey [13] proved that a superposition of k MOLS of order n (with pairwise disjoint symbol-sets) is A-, D- and E-optimal.
- Bailey [2] proved that for all $s \geq 1$, an s-fold inflation of a superposition of $n-1$ MOLS of order n is A-, D- and E-optimal.
- Bailey [2] classified the 3×3 semi-Latin squares and determined the ones that are A-, D- and E-optimal.
- Chigbu [14] determined the $(4 \times 4)/4$ semi-Latin squares that are A-, D- and E-optimal.
- Using the computational methods described in this chapter, Soicher [37] classified the $(4 \times 4)/k$ semi-Latin squares for $k = 5,\dots,10$, and determined those that are A-, D- and E-optimal.

It is widely believed that when a SOMA(k,n) exists, then one that is optimal in the class of all SOMA(k,n)s is in fact optimal in the class of all $(n \times n)/k$ semi-Latin squares. There are not even two MOLS of order 6, but there are SOMA$(2,6)$s and SOMA$(3,6)$s. These have been classified and the best with respect to various measures of efficiency have been determined (see [5, 8, 25, 27, 31, 37]). There is no SOMA$(k,6)$ with $k > 3$. It is not known whether there are three MOLS of order 10, but there are SOMA$(3,10)$s and SOMA$(4,10)$s (see [5, 30, 31]).

1.4 Uniform semi-Latin squares

We now introduce the concept of uniform semi-Latin squares, and some results about such squares from [36]. Uniform semi-Latin squares provide Schur-optimal semi-Latin squares of many sizes for which no optimal semi-Latin square was previously known for any optimality criterion.

An $(n\times n)/k$ semi-Latin square S is *uniform* if every pair of entries of S, not in the same row or column, intersect in a constant number $\mu=\mu(S)$ of points. For example, the semi-Latin square (1.1) is uniform, with $\mu=1$.

Note that if S is a uniform semi-Latin square then an s-fold inflation of S is also uniform, and if S and T are both $n\times n$ uniform semi-Latin squares (with disjoint symbol-sets) then the superposition of S and T is also uniform.

Lemma 1.1 ([36]). *If S is a uniform $(n\times n)/k$ semi-Latin square then*

$$\mu(S)=k/(n-1),$$

and in particular, $n-1$ divides k.

Theorem 1.2 ([36]). *An $(n\times n)/(n-1)$ semi-Latin square S is uniform if and only if S is a superposition of $n-1$ MOLS of order n.*

Uniform semi-Latin squares can be seen as generalising the concept of complete sets of MOLS. Since the μ-fold inflation of a uniform semi-Latin square is uniform, we see that the existence of a uniform $(n\times n)/((n-1)\mu)$ semi-Latin square for all integers $\mu>0$ is equivalent to the existence of a complete set of MOLS of order n, and such a complete set exists when n is a prime power. Although there are not even two MOLS of order 6, the following is proved in [36].

Theorem 1.3 ([36]). *There exist uniform $(6\times 6)/(5\mu)$ semi-Latin squares for all integers $\mu>1$.*

The statistical importance of uniform semi-Latin squares is due to the following:

Theorem 1.4 ([36]). *Suppose that S is a uniform $(n\times n)/k$ semi-Latin square. Then S is Schur-optimal; that is, $\Delta(S)$ is Schur-optimal in the class of underlying block designs of $(n\times n)/k$ semi-Latin squares.*

Proof. We give an outline of the proof. Details can be found in [36].

- The dual S^* of S is a partially balanced incomplete-block design with respect to the L_2-type association scheme (see [3]), so we may easily determine the eigenvalues of the concurrence matrix of S^* (see [39]), and hence the canonical efficiency factors of S^*.
- These canonical efficiency factors are $1-1/(n-1)$, with multiplicity $(n-1)^2$, and 1, with multiplicity $2(n-1)$.
- The dual T^* of any $(n\times n)/k$ semi-Latin square T has at most $(n-1)^2$ canonical efficiency factors not equal to 1.

- It follows that S^* is Schur optimal in the class of duals of $(n \times n)/k$ semi-Latin squares, and so $\Delta(S)$ is Schur optimal in the class of underlying block designs of $(n \times n)/k$ semi-Latin squares. □

1.5 Semi-Latin squares from transitive permutation groups

In [36], a simple construction is given which produces a semi-Latin square SLS(G) from a transitive permutation group G, and in this section we discuss how properties of G determine properties of SLS(G).

Let G be a transitive permutation group on $\{1,\ldots,n\}$, with $n > 1$. For all $i, j \in \{1,\ldots,n\}$, there are exactly $|G|/n$ elements of G mapping i to j (the elements mapping i to j are precisely those in the right coset $G_i g$, where G_i is the stabilizer in G of i and g is any element of G with $ig = j$). Thus G defines a semi-Latin square, as follows. The $(n \times n)/k$ semi-Latin square SLS(G), with $k := |G|/n$, has symbol set G itself, and the symbol g is in the (i, j)-entry of SLS(G) if and only if $ig = j$. For example,

$$\mathrm{SLS}(S_3) = \begin{array}{|ll|ll|ll|} \hline 1 & (23) & (12) & (123) & (13) & (132) \\ \hline (12) & (132) & 1 & (13) & (23) & (123) \\ \hline (13) & (123) & (23) & (132) & 1 & (12) \\ \hline \end{array}.$$

Theorem 1.5 ([36]). *Let G be a transitive permutation group on $\{1,\ldots,n\}$, with $n > 1$, and let $S :=$ SLS(G). Then:*

- *Let H be a transitive subgroup of G of index m. Then S is a superposition of m semi-Latin squares, each isomorphic to SLS(H) (this comes from the partition of G into the right (or left) cosets of H). In particular, if H has order n then S is a superposition of $|G|/n$ isomorphic Latin squares.*
- *G contains a non-identity element with exactly f fixed points if and only if there are two distinct symbols of S which occur together in exactly f entries of S.*
- *G is a Frobenius group (that is, a transitive permutation group in which no non-identity element fixes more than one point) if and only if S is a superposition of MOLS.*
- *$\Delta(S)$ is connected if and only if G has no normal subgroup N satisfying $G_1 \le N \ne G$.*
- *The automorphism group of S (i.e. the group of all isomorphisms from S to S) has structure*
$$(G \times G).((N_{S_n}(G)/G) \times C_2)$$
(in ATLAS [15] notation).
- *G is 2-transitive if and only if S is uniform.*

Proof. See [36] for the proofs of these assertions. We only repeat the (easy) proof of the important last statement. Also see [38].

Suppose G is 2-transitive. Then for every $i, i', j, j' \in \{1, \ldots, n\}$ with $i \neq i'$ and $j \neq j'$, there are precisely $\mu := |G|/(n(n-1))$ elements $g \in G$ with $ig = j$ and $i'g = j'$. Thus, $S(i, j)$ and $S(i', j')$ intersect in exactly these μ elements, and so S is uniform.

Conversely, suppose S is uniform. Then if $i, i', j, j' \in \{1, \ldots, n\}$ with $i \neq i'$ and $j \neq j'$, then $S(i, j)$ and $S(i', j')$ intersect in $\mu := k/(n-1) > 0$ symbols (recall that $n > 1$, $k > 0$), so there is an element of G mapping i to j and i' to j'. Thus G is 2-transitive. $\square$

Using the Classification of Finite Simple Groups, all the finite 2-transitive permutation groups have been classified, and tables of these groups are given in [10]. Each 2-transitive group G gives rise to a uniform semi-Latin square $\mathrm{SLS}(G)$, certain properties of which can be deduced from properties of G. For example, consideration of the groups $PGL_2(q)$ and $PSL_2(q)$, of degree $q+1$, where q is a prime power, yields the following result.

Theorem 1.6 ([36]). *Let q be a prime power. Then there exists a uniform, and hence Schur-optimal, $((q+1) \times (q+1))/(q(q-1))$ semi-Latin square S which is a superposition of isomorphic Latin squares and in which every pair of distinct symbols occur together in at most two entries. Moreover, if q is odd then S is also a superposition of two isomorphic uniform $((q+1) \times (q+1))/(q(q-1)/2)$ semi-Latin squares.*

1.5.1 The canonical efficiency factors of SLS(G)

Let G be a transitive permutation group on $\{1, \ldots, n\}$, with $n > 1$, and let Λ be the concurrence matrix of the underlying block design of $\mathrm{SLS}(G)$. Then Λ is a $|G| \times |G|$ matrix whose rows and columns are indexed by the elements of G and whose (g,h)-entry is the number of fixed points of $g^{-1}h$, which is $\pi(g^{-1}h)$, where π is the permutation character of G. Applying this observation, Martin Liebeck (at the Fifth de Brún Workshop itself) discovered and proved the theorem below. The statement of the theorem and its proof use basic representation theory of finite groups over the complex numbers, such as can be found in [20].

Theorem 1.7 (M. Liebeck). *Let G be a transitive permutation group of degree $n > 1$ with permutation character π. Then the canonical efficiency factors of (the underlying block design of)* $\mathrm{SLS}(G)$ *are*

$$1 - \langle \chi, \pi \rangle / \chi(1),$$

repeated $\chi(1)^2$ times, where χ runs over the non-trivial complex irreducible characters of G.

Proof. Suppose $|G| = nk$, and let $GL_{nk}(\mathbb{C})$ be the group of all invertible $nk \times nk$ matrices over the complex numbers, whose rows and columns are indexed by the elements of G, and let $\rho : G \to GL_{nk}(\mathbb{C})$ be defined by $\rho(x)_{g,h} = 1$ if $gx = h$ and $\rho(x)_{g,h} = 0$ otherwise. In other words, ρ is the right-regular matrix representation of G, with natural G-module $V := \mathbb{C}^{nk}$.

Now let $C_0 = \{1\}, C_1, \ldots, C_d$ be the conjugacy classes of G, with respective representatives $c_0, \ldots, c_d$, let $\chi_0, \ldots, \chi_d$ be the complex irreducible characters of G, with χ_0 the trivial character, and let

$$A_i := \sum_{c \in C_i} \rho(c).$$

Then the matrices A_i commute pairwise, and V decomposes into a direct sum of common eigenspaces $V_0, \ldots, V_d$ of $A_0, \ldots, A_d$, with V_j being a G-submodule of V isomorphic to the direct sum of $\chi_j(1)$ copies of the irreducible G-module with character χ_j. In particular V_j has dimension $\chi_j(1)^2$, and V_j is an eigenspace for A_i with corresponding eigenvalue

$$|C_i|\chi_j(c_i)/\chi_j(1).$$

Note that V_0 is spanned by the all-1 vector.

Now A_i is a $(0,1)$-matrix, with (g,h)-entry equal to 1 if and only if $g^{-1}h \in C_i$, and so

$$\Lambda = \sum_{i=0}^{d} \pi(c_i) A_i.$$

Thus, for $j = 0, \ldots, d$, Λ has eigenvalue

$$\sum_{i=0}^{d} \pi(c_i)|C_i|\chi_j(c_i)/\chi_j(1) = \sum_{g \in G} \chi_j(g)\pi(g^{-1})/\chi_j(1) = |G|\langle \chi_j, \pi \rangle/\chi_j(1),$$

repeated $\chi_j(1)^2$ times, where $\langle\ ,\ \rangle$ denotes the standard inner product of characters (so $\langle \chi_j, \pi \rangle$ is the multiplicity of the irreducible character χ_j as a constituent of the permutation character π).

The scaled information matrix of the underlying block design Δ of $\mathrm{SLS}(G)$ is

$$F(\Delta) := I_{nk} - (nk)^{-1}\Lambda,$$

and so the canonical efficiency factors of $\mathrm{SLS}(G)$ are

$$1 - \langle \chi_j, \pi \rangle/\chi_j(1),$$

repeated $\chi_j(1)^2$ times, for $j = 1, \ldots, d$. □

Thus, the canonical efficiency factors of $\mathrm{SLS}(G)$, and its A-, D- and E-efficiency measures can be determined from the degrees and multiplicities of the irreducible constituents of the permutation character of G. This has been done by Eamonn O'Brien (using Magma [9]) for all transitive permutation groups of degree ≤ 23, and by Soicher (using GAP) for the primitive permutation groups of non-prime-power degree $n \leq 500$ and with order $\leq n(n-1)$.

As an illustrative example, let G be the group A_5 in its primitive permutation representation of degree 10. The permutation character of G (in ATLAS notation [15]) decomposes as $1a+4a+5a$, so the canonical efficiency factors of $\mathrm{SLS}(G)$ are 3/4 (with multiplicity 16), 4/5 (with multiplicity 25), and 1 (with multiplicity 18). If six MOLS of order 10 exist, then their superposition S would have canonical efficiency factors 5/6 (with multiplicity 54) and 1 (with multiplicity 5); see [2, Corollary 5.2]. The ratios of the A-, D- and E-efficiency measures of $\mathrm{SLS}(G)$ with those of S are respectively approximately 0.9889, 0.9943 and 0.9, so in the (likely) absence of six MOLS of order 10, $\mathrm{SLS}(G)$ provides a highly efficient (and possibly optimal) $(10\times 10)/6$ semi-Latin square.

1.6 The DESIGN package

GAP [18] is an internationally developed, freely available, Open Source system for Computational Group Theory and related areas in algebra and combinatorics. The DESIGN package [34] is a refereed and officially accepted GAP package which provides functionality for constructing, classifying, partitioning and analysing block designs.

In this section we focus on the DESIGN package functions `BlockDesigns`, used to classify block designs, and `BlockDesignEfficiency`, used to determine efficiency measures of 1-designs. There are many other functions in the DESIGN package to construct and analyse block designs, including isomorphism testing and automorphism group computation, and functions to determine information about t-designs from their parameters. The best reference for all this, and for precise details on the parameters for the functions discussed here, is the DESIGN package documentation, which includes many examples. The details of the techniques used in the DESIGN package can be found in its documented Open Source code.

1.6.1 The `BlockDesigns` function

The most important DESIGN package function is `BlockDesigns`, which can construct and classify block designs satisfying a wide range of user-specified properties. The properties which must be specified are:

- the number v of points (the point set is then $\{1,\ldots,v\}$, although the points may also be given names);
- the possible block sizes;
- for a given t, for each t-subset T of the points, the number of blocks containing T (this number may depend on T).

The properties which may optionally be additionally specified are:

- the maximum multiplicity of a block, for each possible block-size;

- the total number b of blocks;
- the block-size distribution;
- a replication number r (that is, specifying that every point is in exactly r blocks);
- the possible sizes of intersections of pairs of blocks of given sizes;
- a subgroup G of S_v, to specify that G-orbits of block designs are isomorphism classes (default: $G = S_v$, giving the usual notion of isomorphism);
- a subgroup H of G, such that H is required to be a subgroup of the automorphism group of each returned design (default: $H = \{1\}$, but specifying a non-trivial H can be a very powerful constraint; see [16, 21, 32, 33]);
- whether the user wants a single design with the specified properties (if one exists), a list of G-orbit representatives of all such designs (i.e. isomorphism class representatives as determined by G; this is the default), or a list of distinct such designs containing at least one representative from each G-orbit.

The `BlockDesigns` function works by transforming the design classification problem into a problem of classifying cliques with a given vertex-weight sum in a certain graph whose vertices are "weighted" with non-zero vectors of non-negative integers. Each vertex of this graph represents a possible H-orbit of blocks, each with the same specified multiplicity, with two distinct vertices not joined by an edge only when the totality of the blocks they represent cannot be a submultiset of the blocks of a required design. Such a non-edge may be a result of user-specified properties of the required designs, or may be determined by applying block intersection polynomials [12, 33]. The graph problem is then handled by the GRAPE [35] function `CompleteSubgraphsOfGivenSize`, which uses a complicated backtrack search. The reader may wish to consult the reference [21], which gives detailed information on techniques used to classify block designs.

We now give some straightforward examples of the use of the `BlockDesigns` function. A more complicated example will follow in Section 1.7. Note that first we load the DESIGN package, which also loads the GRAPE package for graphs and groups, which is heavily used by the DESIGN package.

```
gap> LoadPackage("design");
---------------------------------------------------------------
Loading  GRAPE 4.5 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~leonard/).
Homepage: http://www.maths.qmul.ac.uk/~leonard/grape/
---------------------------------------------------------------
---------------------------------------------------------------
Loading  DESIGN 1.6 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~leonard/).
Homepage: http://www.designtheory.org/software/gap_design/
---------------------------------------------------------------
true
```

We now classify the 2-$(7,3,1)$ designs.

```
gap> designs:=BlockDesigns(rec( v:=7, blockSizes:=[3],
>    tSubsetStructure:=rec(t:=2, lambdas:=[1] ) ) );
[ rec( autGroup := Group([ (1,2)(5,7), (1,2,3)(5,7,6),
       (1,2,3)(4,7,5), (1,5,3)(2,4,7) ]),
```

```
  blockNumbers := [ 7 ], blockSizes := [ 3 ],
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ],
     [ 2, 4, 7 ], [ 2, 5, 6 ], [ 3, 4, 6 ], [ 3, 5, 7 ] ],
  isBinary := true, isBlockDesign := true, isSimple := true,
  r := 3, tSubsetStructure := rec( lambdas := [ 1 ], t := 2 ),
  v := 7 ) ]
```

There is, as is very well known, just one such design up to isomorphism. Note that GAP has printed the value assigned to the variable `designs`. This output is a list containing exactly one block design, in DESIGN package format, stored as a GAP record with properties stored as record components. This output could have been suppressed by ending the assignment statement with "`;;`" instead of "`;`".

We next classify (but do not display) the 1-$(7,3,3)$ designs having no repeated block and invariant under the group generated by $(1,2)(3,4)$.

```
gap> onedesigns:=BlockDesigns(rec( v:=7, blockSizes:=[3],
>    blockMaxMultiplicities:=[1],
>    requiredAutSubgroup:=Group((1,2)(3,4)),
>    tSubsetStructure:=rec(t:=1, lambdas:=[3] ) ) );;
gap> List(onedesigns,AllTDesignLambdas);
[ [ 7, 3 ], [ 7, 3 ], [ 7, 3 ], [ 7, 3 ], [ 7, 3 ],
   [ 7, 3, 1 ] ]
gap> List(onedesigns,d->Size(AutomorphismGroup(d)));
[ 48, 4, 6, 4, 8, 168 ]
```

For a more serious calculation, used in [24], we classify the blockdesigns having 11 points, such that each block has size 4 or 5, and every pair of distinct points is contained in exactly two blocks. This calculation takes about 220 seconds of CPU-time on a 3.1 GHz PC running Linux.

```
gap> designs:=BlockDesigns(rec(v:=11, blockSizes:=[4,5],
>    tSubsetStructure:=rec(t:=2, lambdas:=[2] ) ) );;
gap> List(designs,BlockSizes);
[ [ 5 ], [ 4, 5 ], [ 4, 5 ], [ 4, 5 ], [ 4, 5 ] ]
gap> List(designs,BlockNumbers);
[ [ 11 ], [ 10, 5 ], [ 10, 5 ], [ 10, 5 ], [ 10, 5 ] ]
gap> List(designs,d->Size(AutomorphismGroup(d)));
[ 660, 6, 8, 12, 120 ]
```

More generally, `BlockDesigns` can construct subdesigns of a given block design, such that the subdesigns each have the same user-specified properties. Here, a *subdesign* of Δ means a block design with the same point set as Δ and whose block multiset is a submultiset of the blocks of Δ. In this case, the default G determining isomorphism is $\mathrm{Aut}(\Delta)$. For example, given a block design Δ having v points and each of whose blocks has size k, we classify the "parallel classes" of Δ by classifying the subdesigns of Δ that are 1-$(v,k,1)$ designs (up to the action of $\mathrm{Aut}(\Delta)$).

A DESIGN package function closely related to `BlockDesigns` is the function `PartitionsIntoBlockDesigns`, which classifies the partitions of (the block multiset of) a given block design Δ, such that the subdesigns of Δ whose block multisets are the parts of this partition each have the same user-specified properties. For example, given a block design Δ having v points and each of whose blocks has size k, we classify the “resolutions” of Δ by classifying the partitions of Δ into 1-$(v,k,1)$ subdesigns (up to the action of $\mathrm{Aut}(\Delta)$).

1.6.2 The `BlockDesignEfficiency` function

To test conjectures and rank designs we have classified, we need to be able to compare efficiency measures **exactly**. We can do this using algebraic computation in GAP, as described in [37], and this functionality is included in the most recent release [34] of the DESIGN package.

If `delta` is a 1-(v,k,r) design (in DESIGN package format) with $v > 1$, and `eps` is a positive rational number, then, in DESIGN 1.6, the function call

```
BlockDesignEfficiency(delta,eps)
```

returns a GAP record `eff` (say) having the following components. The component `eff.A` contains the rational number which is the A-efficiency measure of `delta`, `eff.Dpowered` contains the rational number which is the D-efficiency measure of `delta` raised to the power $v-1$, and `eff.Einterval` is a list $[a,b]$ of non-negative rational numbers such that if E is the E-efficiency measure of `delta` then $a \le E \le b$, $b-a \le$ `eps`, and if E is rational then $a = E = b$. In addition, the component `eff.CEFpolynomial` contains the monic polynomial over the rationals whose zeros (counting multiplicities) are the canonical efficiency factors of the design `delta`.

For example, we calculate the block design efficiency record for one of the 1-$(7,3,3)$ designs classified above.

```
gap> eps:=10^(-8);;
gap> delta:=onedesigns[1];
rec( allTDesignLambdas := [ 7, 3 ],
   autGroup := Group([ (1,2), (6,7), (4,5)(6,7),
      (3,4,5)(6,7), (1,6,2,7)(4,5) ]),
   blockNumbers := [ 7 ], blockSizes := [ 3 ],
   blocks := [ [ 1, 2, 3 ], [ 1, 2, 4 ], [ 1, 2, 5 ],
      [ 3, 4, 5 ], [ 3, 6, 7 ], [ 4, 6, 7 ], [ 5, 6, 7 ] ],
   isBinary := true, isBlockDesign := true,
   isSimple := true, r := 3,
   tSubsetStructure := rec( lambdas := [ 3 ], t := 1 ),
   v := 7 )
gap> eff:=BlockDesignEfficiency(delta,eps);;
gap> eff.A;
21/31
gap> eff.Dpowered;
343/2187
gap> eff.Einterval;
[ 1/3, 1/3 ]
gap> Factors(eff.CEFpolynomial);
[ x_1-1, x_1-1, x_1-7/9, x_1-7/9, x_1-7/9, x_1-1/3 ]
```

1.7 Classifying semi-Latin squares

We now describe how to classify semi-Latin squares via their duals. Our approach, which is somewhat similar to that of Bailey and Chigbu [7], is implemented in the DESIGN package function `SemiLatinSquareDuals`, but can be applied with more flexibility using the function `BlockDesigns`.

The group W_n below will be used to define isomorphism of semi-Latin squares, via their duals. We define

$$W_n := \langle S_n \times S_n, \tau \mid \tau^2 = 1, \tau(a,b)\tau = (b,a) \text{ for all } a,b \in S_n \rangle.$$

Thus W_n is isomorphic to the wreath product $S_n \wr C_2$. If $g \in W_n$ then $g = (a,b)$ or $g = (a,b)\tau$, for some $a,b \in S_n$, and W_n acts on $V := \{1,\ldots,n\}^2$ as follows. For $(i,j) \in V$ and $a,b \in S_n$:

$$(i,j)^{(a,b)} := (i^a, j^b);$$
$$(i,j)^{(a,b)\tau} := (j^b, i^a).$$

Now let S be an $(n \times n)/k$ semi-Latin square, let $S^* = (V, \mathscr{B})$ be the dual of S, and let $g \in W_n$. Define $(S^*)^g := (V, \mathscr{B}^g) = (V, [B^g : B \in \mathscr{B}])$. Then $(S^*)^g$ is the dual of a semi-Latin square T isomorphic to S. Indeed, if $g = (a,b)$ then T can be obtained from S by permuting its rows by a and its columns by b, and if $g = (a,b)\tau$ then T is obtained from S by permuting its rows by a, its columns by b, and then transposing. Conversely, suppose S and T are isomorphic $(n \times n)/k$ semi-Latin squares, with respective duals S^* and T^*. Then T can be obtained from S by applying some row permutation a, some column permutation b, followed possibly by transposing and/or renaming symbols. Then $(S^*)^{(a,b)} = T^*$ if transposing does not take place, and otherwise $(S^*)^{(a,b)\tau} = T^*$. We thus have an action of W_n on the set of duals of $(n \times n)/k$ semi-Latin squares, and the duals of two $(n \times n)/k$ semi-Latin squares X and Y are in the same W_n-orbit if and only if X and Y are isomorphic (as semi-Latin squares); see also [37].

The following GAP function, to be used later, returns a homomorphism from the imprimitive wreath product $S_n \wr C_2$ with block system $\{\{1,\ldots,n\},\{n+1,\ldots,2n\}\}$ onto the group W_n as a permutation group in its action on $V := \{1,\ldots,n\}^2$ as described above. However, the domain of a permutation group in GAP must be a set of positive integers, so the image of the homomorphism is made to be a permutation group on the set $\{1,\ldots,n^2\}$, with i representing the i-th element of V in lexicographic order.

```
gap> L2ActionHomomorphism := function(n)
> local action,tau,W;
> if not IsPosInt(n) then
>    Error("usage: L2ActionHomomorphism( <PosInt> )");
> fi;
> action := function(x,g)
>    # the function which determines the image of
>    #  x  (in {1,...,n^2})  under  g.
```

```
>    local i,j,ii,jj;
>    i:=QuoInt(x-1,n)+1;
>    j:=x-(i-1)*n+n;
>    ii:=i^g;
>    jj:=j^g;
>    if ii<=n then
>      return n*(ii-1)+(jj-n);
>    else
>      return n*(jj-1)+(ii-n);
>    fi;
>    end;
> tau:=Product(List([1..n],i->(i,i+n)));
> W:=Group(Concatenation(
>       GeneratorsOfGroup(SymmetricGroup([1..n])),[tau]));
> # so  W = Sn wr C2  in its imprimitive action on  [1..2*n].
> return ActionHomomorphism(
>           W,            # group
>           [1..n^2],     # domain of action
>           action);      # action of group on domain
> end;;
```

We now define a W_n-invariant block design $U_{n,m} = (V, \mathscr{B}_{n,m})$, which contains the dual of every $(n\times n)/k$ semi-Latin square as a subdesign, as long as this dual has no block of multiplicity greater than m. As before, $V := \{1,\ldots,n\}^2$. The block multiset $\mathscr{B}_{n,m}$ consists of all the subsets of V of the form

$$\{(1,1^g),\ldots,(n,n^g)\},$$

such that $g \in S_n$, and with each such block having multiplicity m. (Note that $U_{n,m}$ is the dual of a certain $(n\times n)/(m(n-1)!)$ semi-Latin square.) The GAP function defined below returns this “universal semi-Latin square dual” $U_{n,m}$ in DESIGN package format.

```
gap> UniversalSemiLatinSquareDual := function(n,m)
> local g,i,block,blocks,U;
> if n<=1 or m<=0 then
>    Error("<n> must be > 1 and <m> must be > 0");
> fi;
> blocks:=[];
> for g in SymmetricGroup([1..n]) do
>    block:=[];
>    for i in [1..n] do
>       Add(block,(i-1)*n+i^g);
>    od;
>    for i in [1..m] do
>       Add(blocks,block);
>    od;
> od;
> U:=BlockDesign(n^2,blocks);
> U.pointNames:=Immutable(Cartesian([1..n],[1..n]));
> return U;
> end;;
```

Observe that a block design $\Delta = (V,\mathscr{B})$ is the dual of an $(n \times n)/k$ semi-Latin square if and only if Δ is a 1-(n^2,n,k) design as well as a subdesign of $U_{n,k}$ (i.e. Δ and $U_{n,k}$ have the same point set and $\mathscr{B}$ is a submultiset of $\mathscr{B}_{n,k}$). We thus obtain the following:

Theorem 1.8 ([37]). *The isomorphism classes of the $(n \times n)/k$ semi-Latin squares are in one-to-one correspondence with the W_n-orbits of 1-(n^2,n,k) subdesigns of $U_{n,k}$. Representatives of these orbits give the duals of isomorphism class representatives of the $(n \times n)/k$ semi-Latin squares.*

This theorem can be adapted to be able to apply the function `BlockDesigns` in the DESIGN package to construct and classify semi-Latin squares whose duals satisfy certain W_n-invariant properties, and/or whose duals are invariant under a user-specified subgroup of W_n. In particular:

Theorem 1.9 ([37]). *The isomorphism classes of the SOMA(k,n)s are in one-to-one correspondence with the W_n-orbits of the 1-(n^2,n,k) subdesigns of $U_{n,1}$ having the property that each pair of distinct blocks meet in at most one point. Representatives of these orbits give the duals of isomorphism class representatives of the SOMA(k,n)s.*

As an application, we find that, up to isomorphism, there are just 2799 SOMA$(2,6)$s, and just 4 SOMA$(3,6)$s (see [31]).

Theorem 1.10 ([37]). *Suppose $n-1$ divides k and let $\mu := k/(n-1)$. The isomorphism classes of the uniform $(n \times n)/k$ semi-Latin squares are in one-to-one correspondence with the W_n-orbits of the subdesigns of $U_{n,\mu}$ with the property that any two points having no co-ordinate in common occur together in exactly μ blocks. Representatives of these orbits give the duals of isomorphism class representatives of the uniform $(n \times n)/k$ semi-Latin squares.*

As an application, we find that, up to isomorphism, there are just 10 uniform $(5 \times 5)/8$ semi-Latin squares, and just 277 uniform $(5 \times 5)/12$ semi-Latin squares.

A further, more complicated application is given below. We use the function `BlockDesigns` to classify, up to isomorphism, the uniform $(6 \times 6)/10$ semi-Latin squares with the property that any two symbols occur together in at most two entries (equivalently, any two blocks in the dual of such a square meet in at most two points). It turns out that there are exactly 98 such semi-Latin squares, and we compute a list L of their duals. We then determine the sizes of the automorphism groups of the elements of L. (The automorphism group of the dual S^* of an $(n \times n)/k$ semi-Latin square S is defined to be the subgroup of W_n that preserves the block multiset of S^*. This is the intersection of W_n with the standard automorphism group of S^* regarded as a block design with no structure on the point set.) The total time taken for this calculation is about seven minutes on a 3.1 GHz PC running Linux.

```
gap> n:=6;;
gap> hom:=L2ActionHomomorphism(n);;
gap> W:=Image(hom);;
gap> mu:=2;;
gap> U:=UniversalSemiLatinSquareDual(n,mu);;
```

```
gap> rel1:=Set(Orbit(W,[1,2],OnSets));;
gap> rel2:=Difference(Combinations([1..n^2],2),rel1);;
gap> L:=BlockDesigns(rec(v:=n^2,
>    blockDesign:=U,  # we are looking at subdesigns of U
>    blockSizes:=[n],
>    tSubsetStructure:=rec(t:=2, partition:=[rel1,rel2],
>       lambdas:=[0,mu]),
>    blockIntersectionNumbers:=[[[0,1,2]]],
>    isoGroup:=W) );;
gap> A:=List(L,x->Intersection(W,AutomorphismGroup(x)));;
gap> autsizes:=List(A,Size);;
gap> Collected(autsizes);
[ [ 4, 24 ], [ 8, 33 ], [ 12, 4 ], [ 16, 14 ], [ 24, 11 ],
  [ 32, 2 ], [ 48, 2 ], [ 80, 1 ], [ 96, 2 ], [ 144, 1 ],
  [ 192, 2 ], [ 576, 1 ], [ 14400, 1 ] ]
```

A further calculation, taking just one second, shows that no design in L is resolvable; equivalently, no semi-Latin square whose dual is in L is the superposition of Latin squares.

```
gap> R:=List(L,design->PartitionsIntoBlockDesigns(rec(v:=n^2,
>    blockDesign:=design,
>    blockSizes:=[n],
>    tSubsetStructure:=rec(t:=1, lambdas:=[1]),
>    isoLevel:=0) ) );;
gap> Collected(List(R,Length));
[ [ 0, 98 ] ]
```

1.8 Efficient semi-Latin squares as subsquares of uniform semi-Latin squares

For n a prime power, Bailey [2] gives a construction which produces efficient (although not necessarily optimal [37]) $(n \times n)/k$ semi-Latin squares for all $k > 1$. Efficient (but not known to be optimal) $(6 \times 6)/k$ semi-Latin squares are known for $k = 2,3,4,5,6$; see [5, 8, 31, 37]. Any uniform $(6 \times 6)/10$ semi-Latin square, such as $\mathrm{SLS}(PSL_2(5))$, is Schur-optimal, and so is A-, D- and E-optimal. This leaves open the problem of finding efficient $(6 \times 6)/k$ semi-Latin squares, for $k = 7,8,9$. To do this, we look at "subsquares" of a uniform $(6 \times 6)/10$ semi-Latin square.

We say that an $n \times n$ semi-Latin square S is a *subsquare* of an $n \times n$ semi-Latin square T if $S = T$ or T is the superposition of S and another $n \times n$ semi-Latin square. Another way of looking at subsquares is via duals. We have S a subsquare of T if and only if the symbol set of S is subset of the symbol set of T and the dual S^* of S is a 1-(n^2,n,r) subdesign of T^*, for some $r > 0$. In [37], subsquares of uniform semi-Latin squares are investigated, and the following result is proved.

Theorem 1.11 ([37]). *Let $n \geq 3$ and let S be an $(n \times n)/k$ subsquare of a uniform $(n \times n)/t$ semi-Latin square T, such that $t - k < n - 1$. Then*

$$\mathrm{E}_S = 1 - t/(k(n-1)) = 1 - \mu(T)/k.$$

Now let $k \in \{7,8,9\}$. By Theorem 1.11, each $(6\times 6)/k$ subsquare of a uniform $(6\times 6)/10$ semi-Latin square has E-efficiency measure $1-2/k$. We obtain an efficient $(6\times 6)/k$ semi-Latin square Y_k by taking the most A-efficient subsquare of that size of a certain uniform $(6\times 6)/10$ semi-Latin square Y_{10}. The square Y_{10} was chosen as follows. The list L above of the duals of the 98 uniform $(6\times 6)/10$ semi-Latin squares with the property that any two distinct symbols occur together in at most two blocks includes the dual of $\mathrm{SLS}(PSL_2(5))$, whose automorphism group has size 14400. However, $\mathrm{SLS}(PSL_2(5))$ has no $(6\times 6)/9$ subsquare (equivalently it has no Latin square of order 6 as a subsquare). The next largest automorphism group size amongst the 98 dual squares in L is 576, and the one square whose dual in L has an automorphism group of size 576 is chosen as Y_{10}, which does indeed have $(6\times 6)/k$ subsquares, for $k=7,8,9$. We have chosen a square whose dual has a large automorphism group to facilitate the classification of subsquares.

```
gap> f:=First([1..Length(L)],i->Size(A[i])=576);;
gap> Y10star:=L[f];;
gap> autY10star:=A[f];;
gap> StructureDescription(autY10star);
"((A4 x A4) : C2) : C2"
```

We give the square Y_{10} columnwise below:

1	2	3	4	5	6	7	8	9	10
11	12	21	22	31	32	41	42	51	52
13	19	23	25	35	37	47	49	53	59
14	16	27	29	33	40	44	46	57	60
15	18	28	30	34	39	43	45	55	58
17	20	24	26	36	38	48	50	54	56

11	12	13	14	15	16	17	18	19	20
1	2	23	24	33	34	43	44	53	54
3	9	21	27	38	39	45	48	51	57
5	8	22	30	32	36	47	50	55	59
6	7	26	29	31	35	42	49	56	60
4	10	25	28	37	40	41	46	52	58

21	22	23	24	25	26	27	28	29	30
3	4	13	14	35	36	45	46	55	56
1	5	11	17	31	40	43	50	58	60
2	7	15	19	37	39	42	48	52	54
8	10	16	20	33	38	41	47	51	53
6	9	12	18	32	34	44	49	57	59

31	32	33	34	35	36	37	38	39	40
5	6	15	16	25	26	47	48	57	58
4	7	18	20	22	29	41	44	54	55
1	10	12	17	23	28	45	49	51	56
2	9	11	13	24	27	46	50	52	59
3	8	14	19	21	30	42	43	53	60

41	42	43	44	45	46	47	48	49	50
7	8	17	18	27	28	37	38	59	60
6	10	14	15	24	30	32	33	52	56
4	9	11	20	21	26	34	35	53	58
1	3	12	19	22	25	36	40	54	57
2	5	13	16	23	29	31	39	51	55

51	52	53	54	55	56	57	58	59	60
9	10	19	20	29	30	39	40	49	50
2	8	12	16	26	28	34	36	42	46
3	6	13	18	24	25	31	38	41	43
4	5	14	17	21	23	32	37	44	48
1	7	11	15	22	27	33	35	45	47

We now classify the $(6\times 6)/7$ subsquares of Y_{10} by classifying the 1-$(36,6,7)$ subdesigns of Y_{10}^*, up to the action of the automorphism group of Y_{10}^*, as follows.

```
gap> k:=7;;
gap> subdesigns:=BlockDesigns(rec(v:=n^2,
>     blockDesign:=Y10star,
>     blockSizes:=[n],
>     tSubsetStructure:=rec(t:=1, lambdas:=[k]),
>     isoGroup:=autY10star) );;
gap> Length(subdesigns);
150
```

The determination of these 150 subdesigns takes about seven minutes on a 3.1 GHz PC running Linux.

We next determine the design(s) in the list `subdesigns` with the highest A-efficiency measure. There is just one such subdesign, and it also has the highest D-efficiency measure of those in the list. These calculations take about 16 seconds.

```
gap> eff:=List(subdesigns,BlockDesignEfficiency);;
gap> maxA:=Maximum(List(eff,x->x.A));
18972014997910099125/22524377910796536046
gap> pos:=Filtered([1..Length(eff)],j->eff[j].A=maxA);
[ 65 ]
gap> ForAll([1..Length(eff)],
>     j->eff[pos[1]].Dpowered>=eff[j].Dpowered);
true
```

Let Y_7 be the subsquare of Y_{10} whose dual Y_7^* is the design in the list `subdesigns` with the highest A-efficiency measure. Then Y_7 is the $(6\times 6)/7$ semi-Latin square obtained from Y_{10} by removing the $(6\times 6)/3$ semi-Latin square induced on the symbols

$$\{4,8,9,13,15,17,22,25,26,32,33,39,43,46,49,51,54,60\}.$$

The A-efficiency measure of Y_7 is

$$22224360426123258975/25776723339009695896 \approx 0.8622.$$

Similar calculations find a $(6\times 6)/8$ subsquare of Y_{10} with the highest A-efficiency measure. It also has the highest D-efficiency measure. This subsquare Y_8 can be obtained from Y_{10} by removing the $\mathrm{SOMA}(2,6)$ induced on the symbols

$$\{2,6,17,19,21,29,33,36,41,45,58,59\}.$$

The A-efficiency measure of Y_8 is $3643863/4141988 \approx 0.8797$.

Up to isomorphism, we find there is just one $(6\times 6)/9$ subsquare of Y_{10}. We may take this to be the semi-Latin square Y_9 obtained from Y_{10} by removing the Latin square induced on the symbols

$$\{8,11,25,39,44,56\}.$$

The A-efficiency measure of Y_9 is $2968/3323 \approx 0.8932$.

1.9 Some open problems

We conclude this chapter with some open problems.

When n is a prime power or $n=6$, we know precisely the values of μ for which there exists a uniform $(n\times n)/(\mu(n-1))$ semi-Latin square, but we do not know exactly which values of μ have this property for any other $n>1$. The first unsettled case is $n=10$. It is a celebrated computational result that there is no projective plane of order 10 [22, 28], so there do not exist nine MOLS of order 10, and so a uniform $(10\times 10)/9$ semi-Latin square does not exist. On the other hand, $\mathrm{SLS}(PSL_2(9))$ and inflations of this square yield uniform $(10\times 10)/(9\mu)$ semi-Latin squares for $\mu = 4,8,12,16,\ldots$. Our first question is: do there exist uniform $(10\times 10)/18$ or $(10\times 10)/27$ semi-Latin squares?

We have classified certain types of uniform $(6\times 6)/10$ semi-Latin squares, and found none which is a superposition of Latin squares. Our second question is: does there exist a uniform $(6\times 6)/10$ semi-Latin square which is a superposition of ten Latin squares?

Finally, are there general constructions for optimal (say E-optimal) $(n\times n)/k$ semi-Latin squares for when there do not exist k MOLS of order n and there is no uniform $(n\times n)/k$ semi-Latin square? For example, is every $(n\times n)/(k-1)$ subsquare of a uniform $(n\times n)/k$ semi-Latin square E-optimal?

Acknowledgements I am grateful to Martin Liebeck for his result of Section 1.5.1 and for allowing its inclusion in this chapter. I also thank Eamonn O'Brien for his calculations in Magma.

The hospitality of the de Brún Centre for Computational Algebra, NUI Galway, during the Fifth de Brún Workshop is gratefully acknowledged.

References

1. R.A. Bailey, Semi-Latin squares, *J. Statist. Plann. Inference* **18** (1988), 299–312.
2. R.A. Bailey, Efficient semi-Latin squares, *Statist. Sinica* **2** (1992), 413–437.
3. R.A. Bailey, *Association Schemes. Designed Experiments, Algebra and Combinatorics*, Cambridge University Press, Cambridge, 2004.
4. R.A. Bailey, *Design of Comparative Experiments*, Cambridge University Press, Cambridge, 2008.
5. R.A. Bailey, Symmetric factorial designs in blocks, *J. Stat. Theory Pract.* **5** (2011), 13–24.
6. R.A. Bailey and P.J. Cameron, Combinatorics of optimal designs, in *Surveys in Combinatorics 2009*, S. Huczynska et al. (eds), Cambridge University Press, Cambridge, 2009, pp. 19–73.
7. R.A. Bailey and P.E. Chigbu, Enumeration of semi-Latin squares, *Discrete Math.* **167-168** (1997), 73–84.
8. R.A. Bailey and G. Royle, Optimal semi-Latin squares with side six and block size two, *Proc. Roy. Soc. London Ser. A* **453** (1997), 1903–1914.
9. W. Bosma, J. Cannon and C. Playoust, The Magma algebra system I. The user language, *J. Symbolic Comput.* **24** (1997), 235–265.
10. P.J. Cameron, *Permutation Groups*, Cambridge University Press, Cambridge, 1999.
11. P.J. Cameron, P. Dobcsányi, J.P. Morgan and L.H. Soicher, The External Representation of Block Designs, `http://designtheory.org/library/extrep/extrep-1.1-html/`
12. P.J. Cameron and L.H. Soicher, Block intersection polynomials, *Bull. Lond. Math. Soc.* **39** (2007), 559–564.
13. C.-S. Cheng and R.A. Bailey, Optimality of some two-associate-class partially balanced incomplete-block designs, *Ann. Statist.* **19** (1991), 1667–1671.
14. P.E. Chigbu, Semi-Latin Squares: Methods for Enumeration and Comparison, PhD Thesis, University of London, 1996.
15. J.H. Conway, R.T. Curtis, S.P. Norton, R.A. Parker and R.A. Wilson, *ATLAS of Finite Groups*, Clarendon Press, Oxford, 1985.
16. P. Dobcsányi, D.A. Preece and L.H. Soicher, On balanced incomplete-block designs with repeated blocks, *European J. Combin.* **28** (2007), 1955–1970.
17. R.N. Edmondson, Trojan square and incomplete Trojan square designs for crop research, *J. Agricultural Science* **131** (1998), 135–142.
18. The GAP Group, GAP — Groups, Algorithms, and Programming, Version 4.5.2(beta), 2011, `http://www.gap-system.org/`
19. A. Giovagnoli and H.P. Wynn, Optimum continuous block designs, *Proc. Roy. Soc. London Ser. A* **377** (1981), 405–416.
20. G. James and M. Liebeck, *Representations and Characters of Groups*, Second Edition, Cambridge University Press, Cambridge, 2001.
21. P. Kaski and P.R.J. Östergård, *Classification Algorithms for Codes and Designs*, Springer, Berlin-Heidelberg-New York, 2006.
22. C.W.H. Lam, L. Thiel and S. Swiercz, The non-existence of finite projective planes of order 10, *Canad. J. Math.* **41** (1989), 1117–1123.
23. J.H. van Lint and R.M. Wilson, *A Course in Combinatorics*, Cambridge University Press, Cambridge, 1992.
24. J.P. McSorley and L.H. Soicher, Constructing t-designs from t-wise balanced designs, *European J. Combin.* **28** (2007), 567–571.
25. N.C.K. Phillips and W.D. Wallis, All solutions to a tournament problem, *Congr. Numerantium* **114** (1996), 193–196.

26. D.A. Preece and G.H. Freeman, Semi-Latin squares and related designs, *J. Roy. Statist. Soc. Ser. B* **45** (1983), 267–277.
27. D.A. Preece and N.C.K. Phillips, Euler at the bowling green, *Util. Math.* **61** (2002), 129–165.
28. D.J. Roy, Confirmation of the Non-existence of a Projective Plane of Order 10, MSc Thesis, Carleton University, Ottawa, 2011.
29. K.R. Shah and B.K. Sinha, *Theory of Optimal Designs*, Lecture Notes in Statistics **54**, Springer, Berlin-Heidelberg-New York, 1989.
30. L.H. Soicher, On the structure and classification of SOMAs: generalizations of mutually orthogonal Latin squares, *Electron. J. Combin.* **6** (1999), R32, 15 pp. Printed version: *J. Combin.* **6** (1999), 427–441.
31. L.H. Soicher, SOMA Update, `http://www.maths.qmul.ac.uk/~leonard/soma/`
32. L.H. Soicher, Computational group theory problems arising from computational design theory, *Oberwolfach Rep.* **3** (2006), Report 30/2006: Computational group theory, 1809–1811.
33. L.H. Soicher, More on block intersection polynomials and new applications to graphs and block designs, *J. Combin. Theory, Ser. A* **117** (2010), 799–809.
34. L.H. Soicher, The DESIGN package for GAP, Version 1.6, 2011, `http://designtheory.org/software/gap_design/`
35. L.H. Soicher, The GRAPE package for GAP, Version 4.5, 2011, `http://www.maths.qmul.ac.uk/~leonard/grape/`
36. L.H. Soicher, Uniform semi-Latin squares and their Schur-optimality, *J. Combin. Des.* **20** (2012), 265–277.
37. L.H. Soicher, Optimal and efficient semi-Latin squares, submitted for publication. Preprint at: `http://www.maths.qmul.ac.uk/~leonard/statsls.pdf`
38. C.-Y. Suen and I.M. Chakravarti, Efficient two-factor balanced designs, *J. Roy. Statist. Soc. Ser. B* **48** (1986), 107–114.
39. M.N. Vartak, The non-existence of certain PBIB designs, *Ann. Math. Stat.* **30** (1959), 1051–1062.