Pointwise Defining Sets and Trade Cores

Cathy Delaney, Brenton D. Gray, Ken Gray, Barbara M. Maenhaut, Martin J. Sharry and Anne Penfold Street

Centre for Combinatorics, Department of Mathematics
The University of Queensland, Brisbane 4072, AUSTRALIA

Dedicated to the memory of Derrick Breach, 1933–1996

Abstract

A *block design* $D = (V, \mathcal{B})$ is a set V of v elements together with a set $\mathcal{B}$ of b subsets of V called *blocks*, each containing exactly k elements, such that each element of V occurs in precisely r blocks, for some positive integers r and k. D is called a *t-design* if every t-subset of V occurs in exactly λ_t blocks, for some positive integer λ_t. Such a design D is described as a t-(v, k, λ_t) design.

A t-(v, k, λ_t) *defining set* has previously been defined as a set of blocks which is a subset of a unique t-(v, k, λ_t) design. A defining set is now more broadly defined to be a set of full and/or partial blocks which is contained in a unique t-(v, k, λ_t) design. It is a *pointwise* defining set if partial blocks are present. If only full blocks are present, it may be considered as either a pointwise or a *blockwise* defining set.

The results presented here lead to useful tools for finding both pointwise defining sets of designs, and the relevant generalization of trades. Some examples are given to illustrate this.

1 Definitions and Well Known Results

Here a summary is presented of results relevant to finding pointwise defining sets of designs. Many details and proofs regarding blockwise defining sets of designs can be found in K Gray [8], [9], [10]. Summaries of theoretical and practical results on this topic are given in Street [20], [21].

Previous papers on defining sets dealt with blockwise defining sets only. The definitions given here have been broadened so that they are relevant to both blockwise and pointwise defining sets.

Australasian Journal of Combinatorics 16(1997), pp.51–76

A **combinatorial design** is a finite set V and a collection $\mathcal{B}$ of subsets of V called **blocks**, and is denoted $(V, \mathcal{B})$. If at least one of the blocks in $\mathcal{B}$ is a proper subset of V, then the design is called **incomplete**. It is said to be a **block design** if there exist positive integers k and r such that each block contains precisely k elements of V, and each element of V occurs in precisely r blocks.

A block design is said to be a ***t*-design** if there is a positive constant λ_t such that every subset of t elements of V appears in precisely λ_t blocks of $\mathcal{B}$.

For ease of notation in describing t-designs we refer to a set of m elements as an m-set, and in writing blocks we drop the set notation.

Example 1.1 The set of blocks $\{123, 134, 256, 456\}$ on the set $V = \{1, \ldots, 6\}$ forms a block design with $v = 6$, $r = 2$, $b = 4$ and $k = 3$. It is not a 2-design since not all pairs occur the same number of times; for example, 12 occurs once and 56 occurs twice.

If $V = \{1, \ldots, 7\}$ and $\mathcal{B}_1 = \{124, 235, 346, 457, 561, 672, 713\}$, then $F_1 = (V, \mathcal{B}_1)$ is a t-design, where $t = 2$, $v = 7 = b$, $k = 3 = r$ and $\lambda_t = 1$.

A t-design based on v elements with block size k is denoted as a t-(v, k, λ_t) design. When $t = 2$ it is called a **balanced** design, denoted (v, k, λ), and in this case $v \leq b$. When equality holds, that is, when $v = b$, the design is said to be **symmetric**. A design with the property that any two blocks intersect in a constant number of elements is said to be **linked**. Any design which is both balanced and symmetric is also linked, with any two blocks intersecting in λ elements. A t-design with no repeated blocks is said to be **simple**. The designs used as examples throughout this paper are simple, though the results given apply generally unless otherwise specified.

Theorem 1.2 *The following relationships between the parameters of a design must hold:*

(i) *for any block design, $vr = bk$; and*

(ii) *for any t-(v, k, λ_t) design, $\lambda_t\binom{v}{t} = b\binom{k}{t}$.* □

Remark 1.3 Clearly any t-design is also an s-design for $0 \leq s < t$. Using the above equation to express both λ_s and λ_t, and then eliminating b from the equations obtained, gives

$$\lambda_s = \frac{\lambda_t\binom{v-s}{t-s}}{\binom{k-s}{t-s}}.$$

This formula must hold for all $0 \leq s < t$. Note that $\lambda_1 = r$ and $\lambda_0 = b$.

Given the parameters t, v, k and λ_t, these equations can be used to calculate values for b, r and λ_s for all $s < t$. Clearly, for a t-design to exist, each parameter calculated using these formulae must have an integer value.

Definition 1.4 *Let D be a t-(v,k,λ_t) design with blocks $\mathcal{B} = \{B_i \mid i = 1, ..., b\}$. A* **partial design** *S of D is a set of subsets of some of those blocks. That is,*

$$S = \{S_i \mid S_i \subseteq B_i;\ i \in I \subseteq \{1, ..., b\}\ \}.$$

If $|S_i| < k$ for at least one $i \in I$, then S is said to be a **pointwise** *partial design. Otherwise it can be considered as a* **blockwise** *or pointwise partial design. For a blockwise partial design, the* **cardinality** *is defined to be the number of blocks it contains, $|I|$; for a pointwise partial design, the* **cardinality** *is defined to be the total number of points in all its blocks, $\Sigma|S_i|$.*

Definition 1.5 *A* **defining set** *S of a t-(v,k,λ_t) design D is a blockwise or pointwise partial design of D, which is a partial design of no other t-(v,k,λ_t) design.*

That is, S is contained within a unique t-(v,k,λ_t) design D, so given the blocks (partial and full) of S there is precisely one **completion** of S to a t-(v,k,λ_t) design, the design D. S is said to be a **blockwise** defining set if it is a blockwise partial design of D. If S is a pointwise partial design of D, it is called a **pointwise** defining set.

Remark 1.6 Note that if S is a partial design of a design D, and S is not a defining set, then no partial design contained in S is a defining set. This is obvious since any partial design S' of S will be contained in every design containing S.

We now make a general definition which may be applied to either blockwise or pointwise defining sets by using the appropriate interpretation of "partial design" and "cardinality".

Definition 1.7 *Let D be a t-(v,k,λ_t) design. A defining set of D is said to be* **minimal** *if it contains no proper partial design which is also a defining set of D. A* **smallest** *defining set of D is any defining set with minimum cardinality among all defining sets of D. A smallest defining set is necessarily minimal.*

Example 1.8 Consider the design F_1 of Example 1.1, and five other 2-$(7,3,1)$ designs based on the same set V with blocks as follows:

$$\begin{aligned}
\mathcal{B}_2 &= \{124, 235, 347, 456, 571, 672, 613\};\\
\mathcal{B}_3 &= \{124, 135, 167, 237, 256, 346, 457\};\\
\mathcal{B}_4 &= \{127, 134, 156, 235, 246, 367, 457\};\\
\mathcal{B}_5 &= \{126, 137, 145, 235, 247, 346, 567\};\\
\mathcal{B}_6 &= \{126, 134, 157, 235, 247, 367, 456\}.
\end{aligned}$$

(i) No two blocks form a defining set of F_1. For instance, let $R = \{124, 235\}$. These two blocks force the block 672, but the set $R' = \{124, 235, 672\}$ is also contained in the design F_2 and hence is not a defining set. Since $R \subset R'$, R is not a defining set either.

(ii) $S = \{124, 235, 346\}$ is a blockwise defining set of F_1 which is smallest by (i), and hence minimal.

(iii) $S' = \{124, 235, 346, 672\}$ is a blockwise defining set of F_1 which is neither minimal nor smallest, since $S \subset S'$.

(iv) $S'' = \{124, 235, 34, 47\}$ is a pointwise defining set of F_1. To see this, note that 672 is forced as in (i). Now the partial block 47 must be completed with one of 1, 3 or 5. Since the pairs 14 and 34 have already appeared, the block 457 is forced, and the rest of F_1 is easily completed. S'' is not a smallest defining set since it has cardinality 10, whereas S, considered as a pointwise defining set, has cardinality 9.

However S'' is a minimal pointwise defining set, which we can see as follows. Removing any point from either of the two partial blocks of S'' leaves a partial design contained in both F_1 and F_2. Removing any point from the block 235 of S'' leaves a partial design contained in both F_1 and F_3. Removing one of the points 1, 2 or 4 from the block 124 of S'' leaves partial designs contained in F_1, and in F_4, F_5 and F_6 respectively.

Definition 1.9 *Two t-(v, k, λ_t) designs D_1 and D_2, based on the sets V_1 and V_2 respectively, are said to be* **isomorphic** *if there exists a bijection $\rho : V_1 \to V_2$ such that $D_1\rho = \{B_i\rho \mid B_i \text{ is a block of } D_1\} = D_2$.*

Example 1.10 Consider the designs F_1 and F_2 of Example 1.8 and let ρ be the transposition (67). Then $F_1\rho = \{124, 235, 347, 456, 571, 672, 613\} = F_2$. Hence F_1 and F_2 are isomorphic.

Definition 1.11 *An* **automorphism** *of a design D=(V, $\mathcal{B}$) is a permutation on V which maps the collection of blocks $\mathcal{B}$ onto itself.*

The set of all automorphisms of a design D forms a group under composition and is denoted by $Aut(D)$. The number of distinct designs isomorphic to a design D based on v points is determined by the size of the automorphism group and is given by $v!/|Aut(D)|$.

The definitions of an automorphism of a partial design, and of its automorphism group, are similar.

Example 1.12 Again consider F_1 as given in Example 1.8, and now let ρ be the cycle (1234567). Then $F_1\rho = F_1$, so ρ is an automorphism of F_1, that is, $\rho \in Aut(F_1)$. In fact, $Aut(F_1)$ is generated by the following four permutations acting on the points of the design: $\rho_1 = (35)(67)$, $\rho_2 = (36)(57)$, $\rho_3 = (23)(47)$ and $\rho_4 = (12)(57)$.

Lemma 1.13 *Let S be a particular defining set of a t-(v, k, λ_t) design D=(V, $\mathcal{B}$), and let $S\rho$ denote the image of the blocks of S under ρ, a permutation of the elements of V which is an automorphism of the design D. Then:*

(i) *$S\rho$ is also a defining set of D;*

(ii) *$Aut(S)$ is a subgroup of $Aut(D)$;*

(iii) *$Aut(D) = \{\rho : S\rho \subseteq \mathcal{B}\}$.* □

Remark 1.14 This lemma has important implications for the task of finding smallest defining sets. From (i) we know that if S is (or is not) a defining set, then every S' isomorphic to S will also be (or not be) a defining set. That is, being a defining set is a class property for isomorphism classes of partial designs. From (ii) we conclude that if $Aut(S)$ is not a subgroup of $Aut(D)$ then S is not a defining set of D. This leads to the weaker but still useful condition that if $|Aut(S)|$ does not divide $|Aut(D)|$ then S is not a defining set.

Definition 1.15 *Let $1 \leq t < k < v$, and let V be a v-set. Its k-subsets are* **blocks**. *A pair of distinct collections of m blocks each, $T = (T_1, T_2)$, is* **mutually t–balanced** *if each t-subset of V is covered by precisely the same number of blocks of T_1 as of T_2. If T_1 and T_2 are disjoint, then T is a (v, k, t)* **trade of volume** *m. If $m = 0$, the trade is* **void**. *Repeated blocks are allowed. The* **foundation** *of the trade, $found(T)$, is the set of elements of V covered by T_1 and T_2. A trade is said to be* **minimal** *if it has no proper subset which is also a trade.*

Note that various definitions are in use here; see for instance [8], [9], [10] and Hwang [15]. In particular, sometimes the term "trade" refers to the pair (T_1, T_2) and sometimes just to T_1.

A **trade** *of a t-(v, k, λ_t) design is a trade $T = (T_1, T_2)$ such that $T_1 \subseteq \mathcal{B}$.*

Equivalently, T contains a trade of the design D if and only if $S = \mathcal{B} \setminus T$ is not a blockwise defining set of D. Clearly, for any trade T of design D, and permutation $\rho \in Aut(D)$, $T\rho$ is also a trade of D.

Note that a trade is a *blockwise* partial design of the design. The concept of a trade will be extended to pointwise partial designs in Section 3.

Example 1.16 Consider F_1 and R' as given in Example 1.8. Since R' is not a defining set, the set T given by

$$T = \mathcal{B}_1 \setminus R' = \{346, 457, 561, 713\}$$

contains a trade. (In this particular case, it *is* a trade, but we cannot assume this in general.) Further, we can use the automorphisms of F_1 to generate other trades. This is easily done by computer, using *nauty* (McKay [18]) to find the generators of the automorphism group, and then applying them to the known trade(s) in as many combinations as desired.

Lemma 1.17 *Let $D = (V, \mathcal{B})$ be a simple t-(v, k, λ_t) design, and let S be a blockwise partial design of D. Then:*

(i) *S is a blockwise defining set of D if and only if S contains a block of every minimal trade T in D;*

(ii) *If $T_1 \subseteq \mathcal{B}$ contains a block of every minimal defining set of D, then $T = (T_1, T_2)$ contains a minimal trade.*

Proof. (i) First suppose S is a blockwise defining set of D.

Let $T_1 \subseteq \mathcal{B}$ be any minimal trade of D, and let T_2 be another set of blocks containing exactly the same t-sets. If S does not contain any block of T_1 then $S \subseteq (\mathcal{B} \setminus T_1)$. Hence $S \subseteq \mathcal{B}$ and also $S \subseteq T_2 \cup (D \setminus T_1)$, another t-(v, k, λ_t) design, contradicting our assumption that S is a defining set.

Now suppose that S contains a block of every minimal trade T in D.

Suppose S is not a defining set of D, so $S \subseteq D'$ for some t-(v, k, λ_t) design $D' = (V, \mathcal{B}')$ distinct from D. Then $T = (\mathcal{B} \setminus S)$ is a trade, since it must contain the same t-sets as $T' = (\mathcal{B}' \setminus S)$, and T is disjoint from S. Also, T contains a minimal trade which must also be disjoint from S, contradicting our assumption that S contains a block of every minimal trade.

(ii) Since $\mathcal{B} \setminus T_1$ contains no defining set of D, it can be completed in at least two ways to designs with the same parameters as D : first, to D itself by taking $(\mathcal{B} \setminus T_1) \cup T_1$; secondly, to D' by taking $(\mathcal{B} \setminus T_1) \cup T_2 = \mathcal{B}'$. But now T_1 and T_2 contain the same t-tuples, and $T = (T_1, T_2)$ contains a trade, and hence a minimal trade. □

Definition 1.18 *A design is said to be* **single-transposition-free** *(STF) if its automorphism group contains no single transposition $(i\ j)$ of two of its elements.*

It is easily verified that any t-$(v, k, 1)$ design with $t < k$ is STF, and that any symmetric 2-(v, k, λ) design is STF.

Example 1.19 This observation, together with Lemma 1.13 allows us to prove easily the minimality of the pointwise defining set S'' of Example 1.8, in the following way: for each partial design Q formed by removing a single point from S'', we find a single transposition ρ such that $Q\rho \subset F_1$. Then by Lemma 1.13 (iii), $\rho \in Aut(F_1)$, contradicting the fact that F_1 is STF. The points removed from S'' and the corresponding transpositions are shown in Table 1. This table also shows the corresponding information for the minimal pointwise defining set M, of F_1, where $M = \{156, 267, 13, 23\}$. M has the same cardinality as S'' but they are not isomorphic.

Lemma 1.20 *[10] Let S be a blockwise defining set of a t-(v, k, λ_t) STF design. Then:*

(i) *S contains at least $v - 1$ elements; and*

(ii) *if any two elements i and j occur only once each in the blocks of S, they must occur in different blocks.* □

Table 1: Two minimal pointwise defining sets of F_1

S''	Point deleted	1	2	4	2	3	5	3	4	3	7
	Transposition	(16)	(25)	(47)	(57)	(56)	(24)	(14)	(14)	(14)	(67)
M	Point deleted	1	5	6	2	6	7	1	3	2	3
	Transposition	(45)	(45)	(67)	(47)	(56)	(47)	(15)	(34)	(27)	(34)

Lemma 1.21 *[10] Let S be a blockwise defining set of a t-(v, k, λ_t) STF design, with $|S| = s$, and let $k^* = min(k, v-k)$. Then:*

(i) $s \geq \frac{2(v-1)}{k^*+1}$, *or equivalently,* $\frac{s(k-2)}{s-2} + 1 \leq v \leq \frac{s(k+1)}{2} + 1$;

(ii) $k^* \leq 2^{s-1}$;

(iii) $v \leq 2^{s-1} + k^* \leq 2^s$; *and*

(iv) $v \leq \binom{s}{2} + s + 1$. □

These formulae allow calculation of a theoretical lower bound for the cardinality of a smallest blockwise defining set for any STF design. (For information on designs which need not be STF, see B D Gray [7].)

Theorem 1.22 *[10] Let S be a blockwise defining set of a simple t-(v, k, λ_t) STF design D, and let $n(S : D)$ denote the number of configurations of blocks of D isomorphic to S. Then $n(S : D) = \frac{|Aut(D)|}{|Aut(S)|}$.* □

Theorem 1.23 *[10] Let $\{D_i\}$ be a transversal of the set of isomorphism classes of the simple t-(v, k, λ_t) designs on V; that is, $\{D_i\}$ is a set of simple t-(v, k, λ_t) designs on V such that every simple t-(v, k, λ_t) design on V is isomorphic to D_i for precisely one value of i. Let S be a configuration on v or $v-1$ elements satisfying:*

(i) *D_i has precisely $|Aut(D)|/|Aut(S)|$ subsets of blocks isomorphic to S;*

(ii) *any t-(v, k, λ_t) design containing a subset of blocks isomorphic to S is isomorphic to D_i.*

Then S is a blockwise defining set of some design isomorphic to D_i. □

This theorem gives sufficient conditions for S to be a blockwise defining set. Condition (i) is easily tested by computer calculations, but condition (ii) requires information about other t-(v, k, λ_t) designs which is more difficult to obtain. Sometimes this information is already known for the particular parameters in question; for example, if it is known that D is the unique t-(v, k, λ_t) design up to isomorphism, then clearly condition (ii) must hold. Gray and Street [11] demonstrated that when this condition is known to hold, this theorem provides a useful shortcut in the algorithm devised by Greenhill [13], [14] for finding smallest defining sets. One result in [11] has also been proved theoretically by the same authors in [12].

2 Finding Smallest Blockwise Defining Sets

An algorithm for finding all smallest defining sets of a given simple STF design was introduced by Greenhill [14]. It made use of the results of [8], [9], [10], described briefly above, to carry out an "intelligent" exhaustive search for smallest defining sets. The original algorithm, its implementation, and results found by applying it to several designs, are presented in detail in [13]. We give a brief explanation of the algorithm.

First the design D and its parameters are checked to ensure they satisfy Theorem 1.2, and the theoretical lower bound for the size of defining sets, n_0, is calculated. Then automorphism group information about the given design D is found: $|Aut(D)|$ is calculated, and its generators are recorded. If any trades have been given, the generators are used to construct more trades from them. The generators are also used later to construct more trades from any trade found by the algorithm.

With these preparatory steps completed, the search for defining sets of size n commences, starting with $n = n_0$, and incrementing n until a defining set is found. For each value of n, the search proceeds as follows:

(a) all n-sets of blocks of the design are found, and sorted into isomorphism classes (since being a defining set is a class property - Remark 1.14);

(b) each class is tested to ensure that:

  (1) $n(S : D)$ divides $|Aut(D)|$ (Theorem 1.22);

  (2) $n(S : D) = \frac{|Aut(D)|}{|Aut(S)|}$ (Theorem 1.22); and

  (3) S contains a block of every known trade (Lemma 1.17).

If a class passes all these tests, it is said to be **feasible**, and a representative S of that class is completed in all possible ways. If only the design D is found, this is a class of defining sets. Otherwise we know that $D \setminus S$ contains a trade, and it is then used to generate more trades to use in test (3). If condition (ii) of Theorem 1.23 is known to hold, any feasible class must consist of defining sets so it is not necessary to search for completions. As shown in [11], this can eliminate the need for lengthy computations.

Algorithms for finding blockwise or pointwise defining sets of a design are dependent on efficient methods for completing a partial design; a new algorithm and implementation are given in detail in Chapter 5 of Delaney [1] and a summary (with user's guide) in Delaney [2]. This new implementation has been tested, using the results given in [13] as a cross-check. Further relevant techniques are given in Lawrence [16] and Ramsay [19]. A new implementation of Greenhill's algorithm for finding smallest blockwise defining sets of a given design is given in Chapter 6 of [1], and a summary (with user's guide) in Delaney, Sharry and Street [3].

3 Blockwise Results Translated

Many results which hold for blockwise defining sets, given in the previous section, are also true for pointwise defining sets. This is easily checked by following through the proofs of the original results and considering partial designs with partial blocks. The following are proved as they are relevant to further results and discussion.

Theorem 3.1 *Let P be a defining set of design D. Then:*

(i) *if $\rho \in Aut(D)$ then $P\rho$ is also a defining set of D;*

(ii) *$Aut(P)$ is a subgroup of $Aut(D)$;*

(iii) *if D is STF, then P has at least $v-1$ distinct elements in its blocks.*

Proof. (i) Suppose ρ is an automorphism of D. If P is a defining set of D then $P\rho$ is a defining set of $D\rho$, and we have $D = D\rho$.

(ii) Again suppose ρ is an automorphism of P. Since $P \subseteq D$ we have $P\rho \subseteq D\rho$. Thus $P = P\rho$ is a partial design of both the designs D and $D\rho$. If D is a t-(v, k, λ_t) design then so is $D\rho$ and, since P is a defining set, $D = D\rho$. Hence $\rho \in Aut(D)$.

(iii) Suppose D is an STF design, and P is a pointwise defining set of D using $v-2$ or fewer distinct points. Then choose two points i and j not appearing in $found(P)$; clearly the transposition $(i\,j)$ is an automorphism of P, but not an automorphism of D since D is STF. Hence, by (ii) above, P is not a defining set. □

Theorem 3.2 *Let P be a defining set of a t-(v, k, λ_t) STF design D and let $\mathcal{A}$ denote the set of blocks of P.*

(i) *If D is simple, and $n(P : D)$ denotes the number of configurations of blocks in D isomorphic to P, we have $n(P : D) = \frac{|Aut(D)|}{|Aut(P)|}$;*

(ii) *Suppose the elements i and j occur the same number of times in P. Let*

$$\mathcal{B}_i = \{B \setminus \{i\} \mid B \in \mathcal{A}, i \in B, j \notin B\}$$

and

$$\mathcal{B}_j = \{B \setminus \{j\} \mid B \in \mathcal{A}, j \in B, i \notin B\}.$$

Then $\mathcal{B}_i$ and $\mathcal{B}_j$ are non-empty and $\mathcal{B}_i \neq \mathcal{B}_j$. That is, i and j must occur in separate blocks of P at least once, and these blocks must differ at points other than i and j. In particular, if the elements i and j occur only once each in P, then i and j must occur in different blocks of P.

Proof. (i) The total number of distinct designs on v elements isomorphic to D is $v!/|Aut(D)|$. Since P is a defining set, it has v or $v-1$ elements occurring in its blocks. Whether P is based on either v or $v-1$ elements, the number of distinct subdesigns isomorphic to P is the same, given by

$$\frac{v!}{|Aut(P)|} = \frac{\binom{v}{v-1}(v-1)!}{|Aut(P)|}.$$

Let $\binom{V}{k}^*$ denote the set of all subsets of V with k or fewer elements. Every design isomorphic to D will contain the same number of subdesigns isomorphic to P. Also no subdesign isomorphic to P can appear in more than one t-(v, k, λ_t) design isomorphic to D, nor in any t-(v, k, λ_t) design non-isomorphic to D, as P is a defining set of D. Hence

$$n(P:D) = \frac{n(P:\binom{V}{k}^*)}{n(D:\binom{V}{k}^*)} = \frac{v!}{|Aut(P)|} \div \frac{v!}{|Aut(D)|} = \frac{|Aut(D)|}{|Aut(P)|}.$$

(ii) Suppose i and j occur the same number of times within the blocks of P. Since P is a defining set, $Aut(P)$ is a subgroup of $Aut(D)$, and since D is STF, the transposition $(i\,j)$ is not an automorphism of P.

If i and j occurred together on each occasion, then the transposition $(i\,j)$ would be an automorphism of P, so i and j must occur in separate blocks at least once. Thus the sets $\mathcal{B}_i$ and $\mathcal{B}_j$ are both non-empty. Now if $\mathcal{B}_i = \mathcal{B}_j$, then all the blocks in which they occur separately could be matched in pairs of blocks identical at all points other than i and j. Then again $(i\,j)$ would be an automorphism of P. So the collections of partial blocks $\mathcal{B}_i$ and $\mathcal{B}_j$ must be different. □

4 New Concepts for Pointwise Defining Sets

4.1 Significance

Consider an STF design D with parameters t-(v, k, λ) and blocks $\mathcal{B} = \{B_i \mid i = 1, \ldots, b\}$. Recall that in the blocks $\mathcal{B}$ if $1 \leq s \leq t$ then every s-set must occur precisely λ_s times, where λ_s is given by

$$\begin{aligned} \lambda_t &= \lambda, \\ \lambda_s &= \frac{\lambda_t.\binom{v-s}{t-s}}{\binom{k-s}{t-s}}, \text{ for } 1 \leq s < t, \end{aligned}$$

and that for $s > t$ any s-set may or may not occur in the design (but will certainly occur at most λ_t times).

Thus in attempting to define a design D, it would seem that inclusion of λ_s copies of an s-set may not tell us anything we do not already know if $s \leq t$, but inclusion of any s-set for $s > t$ *does* help to define the design.

Definition 4.1 *Given a design D with blocks $\mathcal{B} = \{B_i \mid i = 1, \ldots, b\}$ and a partial design P satisfying $P = \{P_i \mid i = 1, \ldots, p;\ P_i \subseteq B_i;\ p \leq b\}$, we say that a partial block P_j with $P_j \subseteq B_j$ for some $j \leq p$ is* **significant in** *P if P has fewer completions than $P \backslash \{P_j\}$; that is, if removing P_j from the collection takes P further from being a defining set.*

Similarly, if we choose a partial block $P_j \subseteq B_j$ with $P_j \notin P$ we say that P_j is **significant to** *P if, by adding P_j to P, we obtain a collection which has fewer completions than P.*

We now consider under what conditions such a P_j may be significant.

(i) If $|P_j| > t$, then P_j may or may not be significant.

(ii) If $|P_j| = s \leq t$ and is significant to P, then we must have $|P| = p \geq \lambda_s$ and there must be at least λ_s blocks P_i satisfying $|P_i \cup P_j| \leq k$. Note that this is a necessary but not a sufficient condition.

Example 4.2 The following examples demonstrate the concept of significance using the 2-(7,3,1) design F_7, with blocks $\mathcal{B}_7 = \{123, 145, 167, 246, 257, 347, 356\}$.

1. $P = \{123, 145\}$; the block 167 has size greater than t, but is not significant to P since it is determined by the blocks already present. Equivalently, if $P = \{123, 145, 167\}$, then no single block of P is significant in P, since any two blocks of P force the third.

2. $P = \{123, 145\}$; the block 246 is significant to P since P has two completions but $P \cup \{246\}$ is a defining set.

3. $P = \{123, 14\}$; the block 26 is not significant to P, since we know it must occur, but not in either of the blocks of P (since it would not "fit").

4. $P = \{123, 14\}$; the block 16 is significant to P since adding it to P rules out completions containing the block 146, such as $\{123, 146, 157, 245, 267, 347, 356\}$.

5. $P = \{123, 14, 46\}$; the block 16 is not significant to P, since using it to complete either 14 or 46 would lead to a t-set occurring twice, so no designs can be constructed that way, and any completion of P will also be a completion of $P \cup \{16\}$.

Theorem 4.3 *Let $P = \{P_i \mid i = 1, \ldots, p\}$ be a partial design, with $|P_j| = s < t$ for some j. If P_j is significant in P then the following hold:*

(i) *if $P' = P \backslash \{P_j\}$, then it is possible for the blocks of P' to be completed to include all λ_s copies of P_j;*

(ii) *among the λ_s blocks of P which can be completed to include copies of P_j, there is at least one block, say P_h, such that $P_j \not\subseteq P_h$ and $|P_h \cup P_j| > t$. Furthermore, if $|P_h| = q \leq t$, then P_h can occur λ_q times in blocks completed from P'.*

Proof. Condition (i) follows immediately from the discussion above.

For condition (ii), recall that since P is a valid partial design and $P' = P \setminus \{P_j\}$, at least one of the λ_s blocks of P' which can be completed to contain P_j does not initially contain P_j. Let P_h be such a block, with $|P_h| = q$.

Now if $|P_h \cup P_j| \leq t$, then $P_h \cup P_j$ must occur a fixed number of times in any completion, and thus would not distinguish between completions of P and P'. So $|P_h \cup P_j| > t$.

If $q > t$ then P_h need not occur again, and if it is possible to include all the t-sets in P_h λ times without repeating P_h, (for example if $\lambda = 1$) then we would have a completion of P' containing $P_h \cup P_j$ which could not be the same as any completion of P.

If $q \leq t$ and P_h can not be included λ_q times in completing the blocks of P', then it must appear elsewhere in the completed design. Such a design would also be a completion of P, since P_j has been completed to include P_h, and P_h still occurs in another block outside of P'. That is, P and P' both require $P_h \cup P_j$ and P_h to appear in separate blocks. □

Remark 4.4 In the algorithm presented in Section 5, the partial designs are generated by progressively removing points from a blockwise defining set. Note that for a particular s-set, conditions for significance may not hold in the current partial design but may become true after one or more further points are removed. That is, an s-set that is not significant at one stage may become significant later, so such a partial design cannot be hastily dismissed, nor such an s-set removed.

However, in the case of $s = 1$ we can derive the following weaker but useful conditions which must hold if P_j is to be significant now or after removing more points:

(i) there must be at least $r = \lambda_1$ other blocks in the partial design;

(ii) at least one of the other blocks must contain a t-set disjoint from P_j.

These conditions are used in selecting partial designs for further testing.

Example 4.5 For the 2-(7,3,1) design we have $r = 3$, so for a 1-set to be significant the partial design must have at least four blocks. If we are given the blockwise

defining set $\{123, 145, 167, 246\}$ of the 2-(7, 3, 1) design F_7, we look for partial designs within it using eight points, which may form pointwise defining sets. From the necessary conditions discussed so far, we can deduce the following:
$\{23, 15, 167, 6\}$ is not a pointwise defining set, since 2 and 3 occur only once each and in the same block, but $\{3, 14, 167, 26\}$ might be a pointwise defining set, since 2 and 3 are now in different blocks;
neither $\{2, 45, 67, 246\}$ nor $\{12, 14, 67, 26\}$ is a pointwise defining set, since each contains only five different elements and we need at least $v - 1 = 6$;
$\{1, 145, 167, 2\}$ might be a pointwise defining set, since there are pairs disjoint from each of 1 and 2.

4.2 Trades and Trade Cores

So far the definition and results given for trades are specific to full blocks. We next explore how these concepts might be of use in finding pointwise defining sets.

Definition 4.6 *A* **configuration** *of p points of a design D is a collection of full and/or partial blocks of D consisting of a total of p points from specified blocks. The blocks of the configuration need not all be of the same size.*

Note that this is different from a pointwise partial design in that we know which full block of D each partial block of a configuration comes from. Any s-set for $1 \leq s \leq t$ occurs in λ_s blocks in D, and in a configuration we know which of those blocks is referred to. Essentially, a partial design is an unordered collection of subsets of the blocks of the design, and a configuration is an ordered collection of subsets of the ordered collection of blocks of the design.

Notation If a design D is given by a list of blocks, for example

$$\mathcal{B}_7 = \{123, 145, 167, 246, 257, 347, 356\},$$

then a configuration P would be written by listing the partial blocks in the same order, including $\emptyset$ for any empty partial blocks. For example,

$$\{\emptyset,\ 145,\ 16,\ 26, \emptyset,\ \emptyset,\ \emptyset\}$$

is a configuration of D using a total of seven points in its blocks. This is referred to as a 7-point configuration or simply a 7-configuration.

If a design D were specified by a table of block numbers and blocks, a configuration would be written in the same format, including $\emptyset$ as necessary, or omitting block numbers corresponding to empty partial blocks.

1	2	3	4	5	6	7
$\emptyset$	145	16	26	$\emptyset$	$\emptyset$	$\emptyset$

or

2	3	4
145	16	26

Definition 4.7 *If $P = \{P_i \mid i = 1, \ldots, b; P_i \subseteq B_i\}$ is a configuration of a design D with blocks $\mathcal{B} = \{B_i \mid i = 1, \ldots, b\}$, then its* **complement** $\overline{P}$ *is given by*

$$\overline{P} = \{\overline{P_i} \mid \overline{P_i} = B_i \backslash P_i; i = 1, \ldots, b\}$$

If any P_i are empty then $\overline{P}$ will include some full blocks of D.

Definition 4.8 *A* **trade core** *C of design D is a configuration in D such that $\overline{C}$ is not a pointwise defining set. C is a* **minimal trade core** *if removing any element from any block in C gives C' where C' is not a trade core, that is, $\overline{C'}$ is a pointwise defining set.*

Lemma 4.9 *If C is a trade core of design D, and T is the set of blocks of D corresponding to the non-empty blocks of C, then T contains a trade.*

Proof. Since T is made up of the full blocks of D corresponding to non-empty blocks in C and $\overline{C}$ is not a pointwise defining set, we know that the blocks $\mathcal{B} \setminus T \subseteq \overline{C}$, so $\mathcal{B} \setminus T$ is not a defining set. Hence T contains a trade. □

Lemma 4.10 *If a p-configuration P of design D is a pointwise defining set, then it must intersect every minimal trade core of D.*

Proof. Let P be a pointwise defining set of D, and C any minimal trade core. Suppose P contains no points of C, so $P \subseteq \overline{C}$. By definition we know $\overline{C}$ is not a pointwise defining set. This is a contradiction, since no subset of a non-defining set can be a defining set. □

Remark 4.11 It is important to consider trade cores as configurations rather than partial designs. It is possible that the partial blocks of a trade core could be present in other blocks of the design, but can only be traded when considered as part of the blocks specified by the configuration. That is, it may be possible to fit the partial blocks of the trade core into other blocks of the design and for these same partial blocks configured in that way not to form a trade core.

Example 4.12 In Table 2, C_1 is a trade core since $\overline{C_1}$ can be completed in two ways ($\mathcal{B}_7$ as given, or with 6 and 7 swapped in blocks $4, \ldots, 7$). C_2 is not a trade core since $\overline{C_2}$ is a pointwise defining set (since blocks 1, 2 and 4 are a defining set contained within $\overline{C_2}$).

Lemma 4.13 *If P is a p-configuration of a design D such that P contains at least one point of every trade core, then P is a pointwise defining set.*

Proof. Suppose we have such a P which is not a pointwise defining set, so $\overline{P}$ is a trade core, by definition. By our assumption P must contain a point of every trade core, that is, P contains a point of $\overline{P}$; this is a contradiction. Hence P must be a pointwise defining set. □

Table 2: Example of trade cores and their complements.

	$\mathcal{B}_7$	C_1	$\overline{C_1}$	C_2	$\overline{C_2}$
1	123	$\emptyset$	123	$\emptyset$	123
2	145	$\emptyset$	145	$\emptyset$	145
3	167	$\emptyset$	167	6	17
4	246	6	24	$\emptyset$	246
5	257	7	25	7	25
6	347	7	34	7	34
7	356	6	35	6	35

Remark 4.14 All this is of little use unless we have some way of finding trade cores. Note that a trade T may have many trade cores.

A minimal trade core can be found by matching the blocks of T_1 to the blocks of T_2, where T_1 is a minimal trade which trades with T_2, and then removing from each block of T_1 all the points common with the corresponding block of T_2. Only points common to T_1 and T_2 can be removed, otherwise the resulting configuration will not be a trade core. To see this, consider a block $B_{1,i} \in T_1$ and its matching block $B_{2,i} \in T_2$. Let x be a point such that $x \in B_{1,i}$ but $x \notin B_{2,i}$, and let T_1^x be T_1 with x removed from $B_{1,i}$. Then $x \in \overline{T_1^x}$ is a point distinguishing between the two trades, which are the possible ways of completing the design. Since T_1 is a minimal trade, there is no smaller portion of it which could be replaced by different k-sets. So $\overline{T_1^x}$ is a defining set, and T_1^x is not a trade core.

Depending on the ordering of blocks in T_2, that is, on how they are matched to blocks of T_1, different trade cores will be produced.

Example 4.15 Let T_1 be the collection of blocks 4, 5, 6 and 7 of the design F_7 given in Example 4.12. Then T_1 forms a trade with the set of blocks $T_2 = \{247, 256, 346, 357\}$. Changing the order of the blocks of T_2 and removing the intersecting points from the blocks of T_1 gives rise to different trade cores, as shown in Table 3.

Table 3: Different trade cores for different orders of the blocks.

T_1	order 1	C_1	order 2	C_2
246	247	6	346	2
257	256	7	357	2
347	346	7	247	3
356	357	6	256	3

A matching of blocks which intersect in t or more points gives an ordering of the blocks of the trades which results in a smaller trade core than other orderings. This is useful when generating trade cores, since smaller trade cores give more stringent testing conditions for pointwise defining sets. It is in fact always possible to match the blocks of a trade so that the matched blocks intersect in t or more points.

To see this, construct a bipartite graph from the trade (T_1, T_2) as follows: the points of the graph are the blocks of T_1 and T_2, and an edge connects a pair of points corresponding to blocks B_1 and B_2 precisely when $B_1 \in T_1$, $B_2 \in T_2$, and $|B_1 \cap B_2| \geq t$.

A **perfect matching** M of a graph G is a subset of its edges such that every point of G appears in precisely one edge of M. Clearly, matching blocks of T_1 with blocks of T_2 so that each pair intersects in t or more points is equivalent to finding a perfect matching in the bipartite graph constructed above.

If X is a subset of the points of a graph G, let $\Gamma(X)$ denote the set of all points which are adjacent to at least one point of X. The conditions for the existence of a perfect matching in the bipartite graph are given by the Marriage Theorem in the following form. (For instance, see Lovasz and Plummer [17] for a full discussion of this theorem.)

Theorem 4.16 *A bipartite graph $G = (A, B)$ has a perfect matching if and only if $|A| = |B|$ and for each $X \subseteq A$, $|X| \leq |\Gamma(X)|$.* □

Applying the Marriage Theorem shows that such a matching exists if and only if the blocks of every subset X_1 of T_1 intersect at least $|X_1|$ blocks of T_2 at t or more points. Proving this property will show that the desired matching for trades must always exist.

Lemma 4.17 *Let T_1 be a trade of t-(v, k, λ_t) design D, and let T_2 be a collection of k-sets containing the same t-sets each with the same multiplicity as in T_1. Then it is possible to order the blocks of T_1 and T_2 so that each block of T_1 intersects with the corresponding block of T_2 in t or more points.*

Proof. Let $X_1 \subseteq T_1$ with $|X_1| = m$. Then the total number of t-subsets in the blocks of X_1, counting repeats, is $m\binom{k}{t}$.

Choose $X_2 \subseteq T_2$ such that the blocks of X_2 contain all the t-subsets contained by X_1 (with multiplicities counted), and there are no blocks in X_2 which contain no t-subset in X_1. That is, there are no unnecessary blocks in X_2. Since X_2 contains all $m\binom{k}{t}$ t-subsets contained by X_1, with at most $\binom{k}{t}$ of these appearing in each block, $|X_2| \geq m$.

Consider the bipartite graph constructed from T_1 and T_2 as described above, where each vertex represents a block from T_1 or T_2 and an edge is drawn between two vertices precisely when they represent a block from T_1 and a block from T_2 which intersect in t or more points.

Since each block in X_2 has at least one t-set in common with some block in X_1, the corresponding vertices are neighbours. Thus $X_2 \subseteq \Gamma(X_1)$, giving $|\Gamma(X_1)| \geq |X_2| \geq m = |X_1|$, and by the Marriage Theorem a perfect matching must exist. □

In order to find every possible minimal trade core, we simply try every possible ordering of the k-sets in trade T_2, that is, the k-sets not in our given design, which trade with the blocks in the trade T_1 of the design. Since trades T_1 and T_2 contain the same t-sets, we order the k-sets of T_2 by matching them to blocks of T_1 which share a common t-set. We find all such orderings using a simple recursive (depth first) search [1], [4].

Example 4.18 We find nine trade cores from Example 4.15, as shown in Table 4.

Table 4: Trade cores produced by recursive search

T_1	C_1		C_2		C_3		C_4		C_5		C_6		C_7		C_8		C_9	
246	247	6	247	6	247	6	256	4	256	4	256	4	346	2	346	2	346	2
257	256	7	256	7	357	2	247	5	247	5	357	2	247	5	256	7	357	2
347	346	7	357	4	346	7	346	7	357	4	247	3	357	4	247	3	247	3
356	357	6	346	5	256	3	357	6	346	5	346	5	256	3	357	6	256	3

The corresponding bipartite graph, with points labelled using the actual blocks, illustrates that each of the orderings which gives a trade core is in fact a perfect matching. The graph is shown in Figure 1, with the perfect matching corresponding to trade core C_1 shown in bold lines.

4.3 Numerical Bounds

Recall that for a pointwise defining set P we have $n(P : D) = \frac{|Aut(D)|}{|Aut(P)|}$ (see Theorem 3.2). This result depends on the fact that D is simple. We might hope that a similar relationship would exist between S and P and their automorphism groups, where S is the smallest number of blocks of D which contain the pointwise defining set P. The simplicity of D ensures that any blockwise partial design S can be uniquely positioned within D. However, if P contains any partial blocks which occur in more than one block of the design D, there may actually be more than one such S, and no property analogous to "simple" holds for P within S. Hence the argument used to obtain the above result does not work in this case.

Thus, in order to use the above result it would be necessary to find all possible partial designs in D isomorphic to P. But counting the number of partial designs isomorphic to P contained within a given blockwise defining set S is of no obvious use.

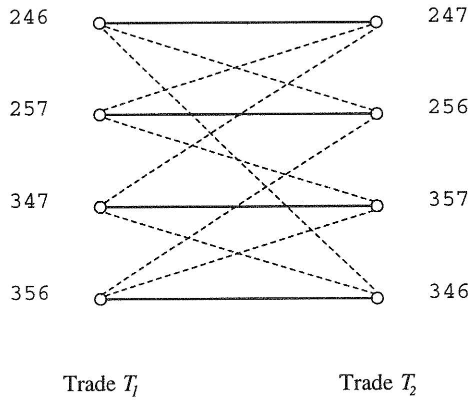

Figure 1: Example of a graph and one perfect matching for trades of the given 2-(7,3,1) design.

Example 4.19 Let $V = \{0, 1, \ldots, 9, a\}$ and let D be the cyclic 2-(11,5,2) design developed from the block 13459, which is the set of quadratic residues (modulo 11). Then

$$S = \{13459, 2456a, 35670, 79a04, 90126\}$$

is a blockwise defining set, and

$$P = \{459, 245a, 3670, 479a0, 1690\}$$

is a pointwise defining set contained in it, found by the algorithm described in the next section. In this case, $n(P : S) = 2$, $|Aut(S)| = 10$ and $|Aut(P)| = 1$, and clearly $n(P : S) \neq \frac{|Aut(S)|}{|Aut(P)|}$.

The algorithm presented in Section 5 does not find $n(P : D)$, but uses the fact that $|Aut(P)|$ must divide $|Aut(D)|$ to rule out partial designs which cannot be pointwise defining sets. Note that since $Aut(P)$ is a subgroup of $Aut(D)$, we could apply more stringent tests to check structures of the groups by looking at the cycles of the generators.

5 Finding Smallest Pointwise Defining Sets

In order to find a smallest pointwise defining set, we first note that any smallest pointwise defining set can be found by removing points from other pointwise defining sets, and ultimately from a blockwise defining set.

Theorem 3.1 and Remark 4.4 provide necessary conditions on the arrangement of elements in a partial design for it to be a defining set. Theorem 3.2 and Lemma 4.10 provide further necessary conditions, relating to the automorphism group order and the trade cores of the design respectively. Hence we can now develop an algorithm which systematically constructs and tests progressively smaller partial designs in search of a smallest pointwise defining set.

The major difficulty in designing a viable algorithm to search for pointwise defining sets is minimizing the number of partial designs to generate and check. In going from blockwise to pointwise defining sets the number of possible partial designs to check increases dramatically. For example, there are $\binom{7}{3} = 35$ ways of choosing three blocks (containing nine points) from the seven blocks of a 2-(7,3,1) design, but there are $\binom{21}{9} = 29\,393$ different 9-configurations in the same design. Note that some of these different configurations will actually represent the same pointwise partial design; for example $\{1, 145, \emptyset, 246, 25, \emptyset, \emptyset\}$ and $\{\emptyset, 145, 1, 246, 25, \emptyset, \emptyset\}$ in design F_7 of Example 4.12.

The algorithm presented here minimizes the number of partial designs to check by starting with known blockwise defining sets and methodically removing points from them. Any pointwise defining set found is kept, and further points removed in search of a smaller pointwise defining set. Partial designs which are not pointwise defining sets are also kept, and used as a checklist for further partial designs generated (since any partial design contained within one which is not a defining set is also not a defining set). This is a form of Tabu search; see for instance Glover [5], [6]. The partial designs are actually generated and stored as configurations, but treated as partial designs when being completed.

Initially, n is set to the number of blocks in a smallest blockwise defining set and p is calculated as $p = nk - 1$.

The three main steps in the algorithm are nested inside each other. They are: finding all n-block defining sets for increasing values of n; removing points from an n-block defining set to find all p-point defining sets within it; removing further points from a p-point defining set to find the smallest defining sets it contains. The value of p is reset and the whole process continues until $n = b$ or $n = p$, that is, until the search has spread the choice of p points across the maximum number of blocks possible. The following is a more detailed description of these three steps.

Step 1: If the list of n-block defining sets is empty, n is incremented and a new list of blockwise defining sets generated. The first n-block defining set is chosen from the list, step 2 is executed, and then this n-block defining set is deleted from the list.

This step is repeated until the search is exhausted, that is, until $n = b$ or $n = p$. This part of the procedure requires the use of a program to find blockwise defining sets, such as that described in [3].

Step 2: Points are methodically removed from the blockwise defining set to ensure that every p-configuration it contains is generated precisely once. Since two configurations may represent identical partial designs, a list of unique partial designs is kept for checking new configurations generated, to avoid unnecessary work in finding the automorphism group information more than once.

The configurations are then checked to see whether they comply with the necessary conditions of Theorems 3.1 and 3.2, Remark 4.4 and Lemma 4.10. If these tests are passed, *complete* [2] is called. If the configuration is found to be a defining set, step 3 is executed, to find the smallest pointwise defining sets within it. The value of p is then set to the number of points in the smallest defining sets found.

This process of generating and testing p-point partial designs within the blockwise defining set continues for decreasing values of p until all configurations have been tested.

Step 3: If a p-point defining set was found by step 2, p is decremented, each point in the defining set found is removed in turn, and the resulting partial designs are grouped by isomorphism. Each class is then tested and kept on a list of defining sets or non-defining sets as appropriate. The non-defining sets are used as another test for possible defining sets, since no partial design contained within a non-defining set can be a defining set.

If any defining sets were found, p is decremented, and a similar search made of each of these new defining sets, producing new lists of defining and non-defining sets.

This is repeated until a value of p is reached for which no defining sets are found.

For each isomorphism class of pointwise partial designs the algorithm must keep a record of a representative configuration and the order of its automorphism group. Since the algorithm generates all configurations from known defining sets, rather than all partial designs from the whole design, no calculation of the size of the classes can be made.

The first version of this algorithm did not use step 2. Instead, all possible p-point configurations within the given blockwise defining set were generated and classified by isomorphism. Then each class was tested, and the search proceeded as in step 3. Generating *all* the p-point configurations was found to take an excessive amount of time and memory, so the new algorithm finds one defining set, searches it for the smallest defining set possible, and then looks for another defining set of the same size to continue the search there. This algorithm terminates more quickly, produces some initial results more rapidly, and uses less memory.

Example 5.1 Consider the 2-(11,5,2) design D of Example 4.19. Let

$$S_1 = \{\mathtt{13459}, \mathtt{2456a}, \mathtt{35670}, \mathtt{79a04}, \mathtt{90126}\}$$

and

$$S_2 = \{\texttt{13459}, \texttt{2456a}, \texttt{35670}, \texttt{46781}, \texttt{57892}\}.$$

Then S_1 and S_2 are representatives of the two isomorphism classes of smallest blockwise defining sets. The smallest pointwise defining sets within these were found using the algorithm outlined above, and the results are outlined in Tables 5 and 6. In summary, the smallest pointwise defining sets within five blocks of this design consist of 20 points; eight non-isomorphic smallest defining sets were found in S_1 and 26 in S_2. The algorithm was not applied to partial designs with six or more blocks, so it is not known whether any defining sets with fewer points spread over more blocks exist.

Comments here relate only to pointwise defining sets within five blocks of the design. In Tables 5 and 6, the columns headed "F_i", for $i = 3, 4, 5$, give the number of blocks in each pointwise defining set which contain i elements. Each defining set has automorphism group of order 1. Note that no 20-point defining set contains only 2 elements in any block, and a minority contain two 3-sets. In fact, of 34 non-isomorphic 20-point defining sets, one comprises five 4-sets, 21 comprise one 3-set, three 4-sets and one 5-set, and 12 comprise two 3-sets, one 4-set and two 5-sets. Thus the points of the 20-point defining sets tend to be fairly evenly spread over the five blocks.

Table 5: 20-Point Defining Sets within S_1.

Points of Defining Set					F_3	F_4	F_5
345	2456a	03567	079	0126	2	1	2
345	245a	0367	0479a	0126	1	3	1
345	256	03567	079a	01269	2	1	2
3459	2456	067	0479a	0126	1	3	1
3459	2456a	0567	49a	0126	1	3	1
3459	256a	356	0479	01269	1	3	1
459	2456	0367	0479a	0126	1	3	1
459	245a	0367	0479a	0169	1	3	1

The algorithm for finding pointwise defining sets of designs depends on the methods for completing a partial design listed in Section 2. The algorithm and its implementation are given in detail in Chapter 7 of Delaney [1], and a summary (with user's guide) is given in Delaney, Maenhaut, Sharry and Street [4].

6 Observations and Open Questions

Several interesting questions arise with regard to finding pointwise defining sets. Is there an optimal partial block size to use when attempting to construct such sets?

Table 6: 20-Point Defining Sets within S_2.

Points of Defining Set					F_3	F_4	F_5
134	245	0567	14678	25789	2	1	2
134	245a	567	14678	25789	2	1	2
1345	245	0357	1468	25789	1	3	1
1345	245a	357	1468	25789	1	3	1
1349	2456	0567	146	25789	1	3	1
1349	2456a	567	146	25789	2	1	2
3459	2456a	567	1678	2589	1	3	1
3459	456	03567	1678	2789	1	3	1
3459	456	0357	1468	25789	1	3	1
3459	456	0567	14678	2589	1	3	1
3459	456	0567	1678	25789	1	3	1
3459	456a	03567	678	2789	1	3	1
3459	456a	0367	678	25789	1	3	1
3459	456a	3567	1678	2789	0	5	0
3459	456a	567	14678	2589	1	3	1
3459	456a	567	1678	25789	1	3	1
3459	46a	03567	678	25789	2	1	2
359	2456	0567	14678	2589	1	3	1
359	2456a	567	14678	2589	2	1	2
359	456	03567	14678	2589	2	1	2
359	456	03567	1468	25789	2	1	2
359	456	0567	14678	25789	2	1	2
359	456a	03567	4678	2789	1	3	1
359	456a	3567	14678	2589	1	3	1
359	456a	567	14678	25789	2	1	2
359	46a	03567	4678	25789	2	1	2

If there is a way of approximating the best block size, does it depend only on the parameters of the design, or is it influenced by other characteristics?

The discussion in this section begins to explore these ideas and others.

Consider a partial block A of size m satisfying $m = \max(t+1, k-t+1)$, so $m > t$ and $k - m < t$. Thus including A is potentially significant in itself, since $m > t$, and we also know that any t-set disjoint from A must occur in a different block, since $m + t > k$; there are $\binom{v-m}{t}$ such t-sets. Perhaps m would be an optimal block size to use in constructing a pointwise defining set of fewest points.

For such m to exist, we know that $k - t \geq 2, t \geq 2$ and $k \geq 4$, since the inequalities above are strict. If such an m were in fact to be an optimal block size for pointwise defining sets, one would expect that a statement akin to the following conjecture would hold.

Conjecture 6.1 *Consider a t-(v, k, λ_t) design with $k - t \geq 2, t \geq 2, k \geq 4$. Then there exists a smallest pointwise defining set of such a design which consists of at most one block of size $< m$, and all other blocks of size $\geq m$, where*

$$m = \max(t+1, k-t+1).$$

Clearly it is not true that *all* smallest pointwise defining sets of such a design must satisfy this condition. The 2-(11,5,2) design satisfies the necessary conditions on the parameters, giving $m = 4$, but we have already seen that it has smallest defining sets which include more than one partial block of only three elements. However, it seems possible that at least one smallest pointwise defining set should satisfy this condition.

If this is shown to be the case, it will provide a useful upper bound on the number of blocks used in the search for smaller and smaller pointwise defining sets. It may also prove to be a useful approximation, and perhaps could lead to a precise necessary condition on the arrangement of points in any pointwise defining set. Note that all the smallest pointwise defining sets of the 2-(11,5,2) design have at most two partial blocks containing only three elements.

It is also interesting to observe characteristics common to all the results obtained so far by applying the algorithm described in the previous section to various designs.

Observation 6.2 *For every* **smallest** *pointwise defining set S that we have found so far for a design D:*
(i) *each partial block of S contains at least $t+1$ elements;*
(ii) *S is a partial design of a smallest blockwise defining set of D, that is, the points are all contained within the blocks of a smallest blockwise defining set of D rather than being scattered over more blocks of the design.*

To discover necessary conditions for these or similar characteristics to be true of **all** smallest pointwise defining sets of a given design could be very useful in streamlining the search-based algorithm of Section 5, and might also lead to development of quick construction-based algorithms. Some theoretical results have been obtained.

Theorem 6.3 *Let D be a t-(v, k, λ_t) design with a smallest pointwise defining set S and a minimal blockwise defining set M, such that each block of S is contained in a block of M. Then each partial block of S contains at least $t + 1$ points.*

Proof. Let the partial blocks of S be $P_1, P_2, \ldots, P_s$ and suppose without loss of generality that $|P_1| = w < t + 1$. Since S is a partial design contained in M, there exists a block B of M such that $P_1 \subset B$. Let $M' = M \setminus B$. and let $M'' = M' \cup P_1$. Since M'' contains the pointwise defining set S as a partial design, M'' itself is a pointwise defining set of D.

We claim that M' is a blockwise defining set of D, and hence that M is not minimal. For the w-set P_1 must occur in some block not in M', and thus M' forces the partial block P_1. In other words, M' forces the defining set M'' and hence M' itself is a defining set. □

But now, by Theorem 6.3, Observation 6.2 (ii) implies Observation 6.2 (i), since a smallest defining set is necessarily minimal. However there is no analogy for **minimal** pointwise defining sets; that is, each partial block of a minimal pointwise defining set need not contain at least $t + 1$ points, nor is it necessarily a partial design of a minimal blockwise defining set.

Example 6.4 In Example 1.19 the minimal pointwise defining sets S'' and M of F_1 are not contained in any minimal blockwise defining set, nor do their partial blocks contain at least $t + 1 = 3$ points.

Acknowledgements

This work has been supported at various times by Australian Research Council grants A49130102 (CD, APS) and A49532477 (BDG, BMM, APS), by Australian Postgraduate Awards (CD, BDG), by a Natural Sciences and Engineering Research Council of Canada PostGraduate Studies Award and a University of Queensland Overseas Postgraduate Research Scholarship (BMM) and by an Australian Senior Research Fellowship (APS).

References

[1] Catherine Margaret Delaney, *Computational aspects of defining sets of combinatorial designs*, M.Sc. Thesis, The University of Queensland, 1995.

[2] Cathy Delaney, *complete* – Rationale and User's Guide, CCRR–01–95, Centre for Combinatorics, Department of Mathematics, The University of Queensland, 1995.

[3] Cathy Delaney, Martin J Sharry and Anne Penfold Street, *bds* – Rationale and User's Guide, CCRR–02–96, Centre for Combinatorics, Department of Mathematics, The University of Queensland, 1996.

[4] Cathy Delaney, Barbara M Maenhaut, Martin J Sharry and Anne Penfold Street, *pds* – Rationale and User's Guide, CCRR–04–96, Centre for Combinatorics, Department of Mathematics, The University of Queensland, 1996.

[5] Fred Glover, *Tabu search – Part I*, ORSA Journal on Computing **1** (Summer 1989), Number 3, 190–206.

[6] Fred Glover, *Tabu search – Part II*, ORSA Journal on Computing **2** (Summer 1989), Number 1, 4–32.

[7] Brenton D Gray, *Defining sets of simple designs*, Bulletin of the Institute of Combinatorics and its Applications **19** (1997), 23-26.

[8] Ken Gray, *On the minimum number of blocks defining a design*, Bulletin of the Australian Mathematical Society **41** (1990), 97-112.

[9] Ken Gray, *Further results on smallest defining sets of well known designs*, Australasian Journal of Combinatorics **1** (1990), 91-100.

[10] Ken Gray, *Defining sets of single-transposition-free designs*, Utilitas Mathematica **38** (1990), 97-103.

[11] Ken Gray and Anne Penfold Street, *Smallest defining sets of the five non-isomorphic* $2-(15,7,3)$ *designs*, Bulletin of the Institute of Combinatorics and its Applications **9** (1993), 96-102.

[12] Ken Gray and Anne Penfold Street, *The smallest defining set of the* $2-(15,7,3)$ *design associated with* $PG(3,2)$*: a theoretical approach*, Bulletin of the Institute of Combinatorics and its Applications **11** (1994), 77-83.

[13] Catherine Suzanne Greenhill, *An algorithm for finding smallest defining sets of t-designs*, M.Sc. Thesis, The University of Queensland, 1992.

[14] Catherine S Greenhill, *An algorithm for finding smallest defining sets of t-designs*, Journal of Combinatorial Mathematics and Combinatorial Computing **14** (1993), 39-60.

[15] H L Hwang, *On the structure of* (v,k,t) *trades*, Journal of Statistical Planning and Inference **13** (1986), 171-191.

[16] Julie L Lawrence, The algorithm *complete* – extensions and improvements, in preparation.

[17] L Lovasz and M D Plummer, *Matching theory*, Annals of Discrete Mathematics **29** (1986), 1–40.

[18] Brendan D McKay, *nauty* User's Guide (Version 1.5), Australian National University Computer Science Technical Report TR-CS-90-02.

[19] Colin Ramsay, An improved version of *complete* for the case $\lambda = 1$, CCRR–03–96, Centre for Combinatorics, Department of Mathematics, The University of Queensland, 1996.

[20] Anne Penfold Street, *Defining sets for block designs: an update*, Combinatorics Advances (edited C J Colbourn and E S Mahmoodian), Kluwer Academic Press, Norwell, Massachusetts, (1995), pp. 307–320.

[21] Anne Penfold Street, *Trades and defining sets,* in CRC Handbook of Combinatorial Designs (edited C J Colbourn and J H Dinitz), CRC Publishing Co, Boca Raton, Florida, (1996), pp. 474–478.

(Received 20/3/97)