Smallest defining sets of some $STS(19)$

PETER ADAMS[1], ABDOLLAH KHODKAR[2] AND COLIN RAMSAY[3]
Centre for Discrete Mathematics and Computing,
The University of Queensland, Brisbane, Qld. 4072, Australia.

Abstract

We describe an algorithm for finding smallest defining sets of designs. Using this algorithm, we show that the 104 $STS(19)$ which have automorphism group order at least 9 have smallest defining set sizes in the range 18–23. The numbers of designs with smallest defining sets of 18, 19, 20, 21, 22 and 23 blocks are, respectively, 1, 2, 17, 68, 14 and 2.

1 Introduction

Let V be a v-set, and suppose that $\mathcal{B}$ is a collection of k-subsets of V with the property that each t-subset of V is in exactly λ of the elements of $\mathcal{B}$. Then the ordered pair $D = (V, \mathcal{B})$ is called a t-(v, k, λ) design. The elements of V are called **points**, and the elements of $\mathcal{B}$ **blocks**. A design with $\lambda = 1$ is called a **Steiner** design, and a 2-$(v, 3, 1)$ design is called a **Steiner triple system** on v points, denoted $STS(v)$.

A set of blocks S which is a subset of a unique t-(v, k, λ) design D is a **defining set** of D. The size of S equals $|S|$ and S is said to be **smallest** if no other defining set of D has smaller size. A defining set is **minimal** if it does not properly contain a defining set. Defining sets were introduced by Gray in the series of papers [3, 4, 5]; see also the survey papers by Street [12, 13].

Let V be a v-set and T_1, T_2 be collections of m k-subsets of V. We say that T_1 and T_2 are t-**balanced** if each t-subset of V is contained in the same number of blocks of T_1 and of T_2. If T_1 and T_2 are disjoint and t-balanced, then $T = \{T_1, T_2\}$ is said to be a (v, k, t) **trade of volume** m.

THEOREM 1: ([3]) *Suppose $D = (V, \mathcal{B})$ and $S \subseteq \mathcal{B}$. Then S is a defining set of D if and only if S intersects each trade in D.*

[1]Department of Mathematics. Research supported by ARC grant A69701550.
[2]Department of Mathematics. Research supported by ARC grant A49937047.
[3]Dept. of Comp. Science & Elect. Engineering. Research supported by the ARC.

JCMCC 38 (2001), pp. 225-230

PROOF: Suppose S is a defining set of D and $\{T_1, T_2\}$ is a trade. If T_1 is a trade in D (that is, $T_1 \subseteq \mathcal{B}$), then $S \cap T_1 \neq \emptyset$, else S is also a subset of the design with blocks $(\mathcal{B} \setminus T_1) \cup T_2$.

Conversely, suppose $S \subseteq \mathcal{B}$ intersects each trade in D. If S is not a defining set of D, then $S \subseteq D_2$ for some design D_2 with the same parameters as, but distinct from, D. Let T_1 comprise the blocks of D not in D_2 and T_2 comprise the blocks of D_2 not in D. Then $\{T_1, T_2\}$ is a trade, with T_1 in D. Since S is disjoint from T_1, this is a contradiction. □

For $v = 7$, 9 and 13, there are one, one, and two non-isomorphic $STS(v)$ respectively. The sizes of smallest defining sets for these were determined in [3, 4, 6], being three, four, and eight and nine respectively. The sizes of smallest defining sets for the 80 non-isomorphic $STS(15)$ were found in [10], and range from eleven to sixteen. In this paper we determine the sizes of smallest defining sets for the $STS(19)$ which have automorphism group orders of at least 9. There are 104 such designs, and we use the listing given in the supplement to [1], labelling the designs, in the order given, #1 to #104.

2 Techniques

We describe two standard techniques for investigating defining sets and trades in designs, and then show how to combine these to yield an algorithm for finding smallest defining sets.

Suppose that $D = (V, \mathcal{B})$ is a design and that $S \subseteq \mathcal{B}$. Then a backtrack search can be used to *complete* S; that is, to find all $STS(19)$ which contain S (see, for example, [10]). If the only completion is D, then S is a defining set and $|S|$ is an upper bound on the size of smallest defining sets. If there is more than one completion, then S is not a defining set and completions not equal to D generate trades in D.

Suppose that $D = (V, \mathcal{B})$ is a design with $\mathcal{B} = \{B_1, B_2, \ldots, B_b\}$, and associate with each block B_j a variable x_j. Given a family of trades $\{T^i\}_{i \in I}$ such that $T^i = \{T^i_1, T^i_2\}$ and $T^i_1 \subseteq \mathcal{B}$ for all $i \in I$, form the inequality $x_{i_1} + x_{i_2} + \cdots + x_{i_s} \geq 1$ for each $T^i_1 = \{B_{i_1}, B_{i_2}, \ldots, B_{i_s}\}$. Now consider the following integer programme:

$$\begin{aligned}
&\text{Minimise } \textstyle\sum_{j=1}^{b} x_j, \text{ subject to} \\
&x_{i_1} + x_{i_2} + \cdots + x_{i_s} \geq 1 \text{ for each } i \in I, \\
&\text{and with } x_j \in \{0,1\} \text{ for all } 1 \leq j \leq b.
\end{aligned}$$

If m is the optimum solution for this integer programming problem, then D has smallest defining set size of at least m. Moreover, if $F = \{x_j \mid 1 \leq j \leq b\}$ is a feasible solution for this system, then $S = \{B_j \mid x_j \in F, x_j = 1\}$ is a smallest defining set of D if it completes uniquely. Of course, even if the lower bound m is tight, S need not have a unique completion. This technique was used in, for example, [8] to find smallest defining sets for the 36 non-isomorphic 2-$(9,3,2)$ designs.

Algorithm: The input is $D = (V, \mathcal{B})$, an $STS(19)$, and the output is a smallest defining set $S \subseteq \mathcal{B}$ of D.

(1) Find some trades of small volume in D and put these in a list $\mathcal{T}$. Typically, these trades will be Pasch trades (that is, the unique $(v,3,2)$ trade of volume four) or trades of volume six. (Only #3, #4, #92, #95, #98 and #100 do not contain any Pasch trades, but they do contain trades of volume six.)

(2) Form the integer programme corresponding to $\mathcal{T}$ and find an optimal solution for this system. Form the set of blocks S corresponding to this solution.

(3) If S has only one completion then stop. Otherwise, there is at least one trade in D which does not intersect S. (Note that two completions of S suffice to show that S is not a defining set; however, the more completions we generate, the more trades we find.)

(4) Use the automorphism group of D to generate all the copies of the trade(s) found in (3) in D, and add these to $\mathcal{T}$.

(5) *Minimise* the list $\mathcal{T}$. That is, if $T_1^x \subseteq \mathcal{B}$ and $T_1^y \subseteq \mathcal{B}$ are distinct trades in D and $T_1^x \subseteq T_1^y$, then delete T_1^y from $\mathcal{T}$.

(6) Go to step (2).

This algorithm should be contrasted with that described by Greenhill [6] which, starting at some lower bound on the size of defining sets, essentially tests all successively larger subsets of the design until a (smallest) defining set is found.

3 Results

To solve the integer programmes we used the CPLEX package [2]. We used *nauty* [9] to obtain sets of generators for the designs' automorphism groups, and GAP [11] to generate the groups' elements from these. The partial-design completion programme for Steiner designs described in [10] was used to check whether solutions to the integer programme were defining sets and to generate trades. The remainder of the work was done using custom-written C programmes.

Our results are summarised in Table 1. Smallest defining sets range from 18 to 23 blocks, and there are 1, 2, 17, 68, 14 and 2 designs with, respectively, sizes of 18, 19, 20, 21, 22 and 23. A text file containing the data in Table 1, along with the blocks of the designs and example smallest defining sets, is available as `www.csee.uq.edu.au/~cram/sts19.txt` on the World-Wide-Web.

As a proportion of the 57 blocks in an $STS(19)$, the sizes of smallest defining sets range from 0.316 to 0.404. In comparison, for the 80 $STS(15)$ the range is 0.314 to 0.457. Interestingly, if the data for the $STS(15)$ which arises from the geometry $PG(3,2)$ is omitted, this range becomes 0.314 to 0.400. Various authors (see, for example, [10, 12]) have noted that, although $|S|$ is not monotonic with either $|A|$ or #P, these values do seem to be related. For the data in Table 1 the coefficient of linear correlation between $|S|$ and $|A|$ is 0.474, and that between $|S|$ and #P is 0.901.

4 Concluding remarks

The results presented here required a great deal of computation. An exact figure is not available but we estimate that, if the time to check the results and the fact that the CPLEX package was run in parallel are taken into account, then approximately one decade of CPU time was required. (The CPUs were mips R10000 units, clocked at 195 MHz.) The bulk of the time was spent solving the optimisation problems, which ranged in size from $\approx$1000 to $\approx$90000 equations. Since the complexity of such problems grows exponentially with the number of variables (that is, blocks), it is unlikely that progress much beyond $v = 19$ for general Steiner triple systems can be made using our technique. However, there are many more $STS(19)$ to which the method could be applied; see, for example, [1] and its bibliography, or [7]. Of course, the algorithm is quite general, and can be applied to other families of designs.

Acknowledgements: We are grateful to the University of Queensland's High Performance Computing Unit for providing computing facilities, and to Anne Street for reading and commenting on an early version of this paper.

TABLE 1: Summary of results. #D is the design's label, $|A|$ is its automorphism group order, #P is the number of Pasch configurations it contains, and $|S|$ is the size of a smallest defining set.

#D	$\|A\|$	#P	$\|S\|$	#D	$\|A\|$	#P	$\|S\|$	#D	$\|A\|$	#P	$\|S\|$
1	432	84	22	36	32	44	21	71	12	66	21
2	108	84	23	37	32	28	21	72	12	66	22
3	171	0	21	38	16	60	22	73	12	60	21
4	57	0	21	39	16	60	22	74	12	60	21
5	57	38	22	40	16	52	21	75	12	60	21
6	144	84	23	41	16	52	21	76	12	60	21
7	96	28	21	42	16	60	21	77	12	66	22
8	24	42	21	43	16	60	22	78	12	66	22
9	24	42	21	44	16	60	22	79	12	15	21
10	24	42	21	45	16	44	22	80	12	7	20
11	24	30	21	46	16	36	21	81	12	14	21
12	24	42	22	47	16	44	21	82	12	22	21
13	24	30	21	48	16	36	21	83	12	18	21
14	24	64	22	49	16	44	21	84	12	18	21
15	24	40	21	50	16	44	21	85	12	23	21
16	24	40	21	51	19	19	21	86	9	48	21
17	24	40	21	52	18	39	18	87	9	21	21
18	24	40	21	53	18	39	19	88	9	15	21
19	12	48	20	54	18	18	21	89	9	15	20
20	12	48	21	55	18	39	21	90	9	15	20
21	12	48	21	56	18	39	21	91	9	15	21
22	12	48	20	57	12	26	21	92	9	0	20
23	12	48	20	58	12	38	21	93	9	9	21
24	12	48	21	59	12	38	21	94	9	18	21
25	12	48	21	60	12	26	21	95	9	0	20
26	12	48	20	61	12	26	21	96	9	18	21
27	54	57	22	62	12	50	21	97	9	36	21
28	54	57	21	63	12	26	21	98	9	0	20
29	18	48	20	64	12	38	21	99	9	27	21
30	18	48	20	65	12	50	21	100	9	0	20
31	18	57	20	66	12	42	21	101	9	9	21
32	18	57	19	67	12	42	21	102	9	18	21
33	18	39	20	68	12	30	20	103	9	18	21
34	18	39	20	69	12	54	22	104	9	36	21
35	32	44	21	70	12	54	21				

References

[1] Charles J. Colbourn, Spyros S. Magliveras and D.R. Stinson. Steiner triple systems of order 19 with nontrivial automorphism group. *Mathematics of Computation*, **59** (1992), 282–295, S25–S27.

[2] CPLEX Callable Library. CPLEX Optimization Inc., 1989–1995.

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

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

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

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

[7] T.S. Griggs and J.P. Murphy, 101 Anti-Pasch Steiner triple systems of order 19, *Journal of Combinatorial Mathematics and Combinatorial Computing*, **13** (1993), 129–141.

[8] A. Khodkar. Smallest defining sets for the 36 non-isomorphic twofold triple systems of order nine. *Journal of Combinatorial Mathematics and Combinatorial Computing*, **17** (1994), 209–215.

[9] Brendan D. McKay, *nauty* User's Guide (Version 1.5). Technical Report TR-CS-90-02, Australian National University, Department of Computer Science, 1990.

[10] Colin Ramsay. An algorithm for completing partials, with an application to the smallest defining sets of the $STS(15)$. *Utilitas Mathematica*, **52** (1997), 205–221.

[11] Martin Schönert et al. GAP – Groups, Algorithms and Programming. Lehrstuhl D für Mathematik, Rheinisch-Westfälische Technische Hochschule, Aachen, 1995.

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

[13] Anne Penfold Street. Trades and defining sets. In: C.J. Colbourn and J.H. Dinitz, Eds., *CRC Handbook of Combinatorial Designs*. CRC Publishing Co., Boca Raton, Florida, 1996, pp. 474–478.