An algorithm for enumerating trades in designs, with an application to defining sets

COLIN RAMSAY[1]
Depts. of Computer Science and of Mathematics,
University of Queensland,
Brisbane, QLD 4072.
(email: cram@cs.uq.edu.au)

Abstract

An algorithm is presented which, when given the non-isomorphic designs with given parameters, generates all the trades in each of the designs. The lists of trades generated by the algorithm were used to find the sizes, previously unknown, of smallest defining sets of the 21 non-isomorphic $2-(10,5,4)$ designs. Consideration of trades in a design to isomorphic and to non-isomorphic designs led to two variations on the concept of defining sets. The lists of trades were then used to find the sizes of these smallest *member* and *class* defining sets, for five parameter sets.

1 Introduction

Given a v-set V, a collection $\mathcal{B}$ of k-subsets (called *blocks*) of V, with the property that each t-subset of V is in exactly λ of the blocks of $\mathcal{B}$, is called a $t-(v,k,\lambda)$ *design*. The number of blocks, $|\mathcal{B}|$, in the design is denoted by b and the number of distinct blocks in $\mathcal{B}$ by b^*. If $b^* = b$ the design is said to be *simple*. We shall be concerned exclusively with simple designs herein. Given a set of parameters for a design, the number of non-isomorphic designs with these parameters is denoted by n. A generic design will be denoted by $D = (V, \mathcal{B})$.

A (v,k,t) *trade* of *volume* s consists of two non-empty disjoint collections, T_1 and T_2, of k-subsets of a v-set V, with $|T_1| = |T_2| = s$, such that for every t-subset of V the number of blocks containing this subset is the same in both T_1 and T_2. It is a standard result that the volume s of a trade is at least 2^t, see [17]. Sometimes we will speak loosely, and say that T_1 (or T_2) is a trade. If $T_1 \subseteq \mathcal{B}$, we say that the design contains the trade. Given a collection of trades in a design, if no proper subset of a trade in the collection is also a trade in the collection, then the trades are said to be *minimal*, and the collection of trades *minimised*.

[1]Research supported by an APRA Scholarship

For some references on trades, see [17, 19, 21]. Trades have many applications in the theory of designs. See, for example, [15, 16, 20] for the use of trades in the construction of designs with different values of b^* and the construction of non-isomorphic designs from a given design.

As a first step toward characterising the trades contained in a $t-(v,k,\lambda)$ design, as opposed to all the (v,k,t) trades with each t-subset of V appearing at most λ times and volume $s \leq b$, an algorithm to enumerate all the trades in the designs with a given set of parameters was developed. This algorithm is presented in Section 2. In Section 3 we show how the lists of trades generated by the algorithm can be used to find the sizes of smallest defining sets of designs. A defining set of a design is a subset of the blocks of a design that can appear in no other design – formal definitions will be given in Section 3. By considering only some of the trades in a design, the technique used in this section leads to the new concepts of *member* and *class* defining sets of designs. In Section 4 we discuss the results obtained using the algorithm, with the results themselves being contained in a series of tables in an appendix at the end of this paper.

2 Algorithm

Given parameters $2 \leq t < k < v$, $\lambda \geq 1$, suppose n is the number of non-isomorphic $t-(v,k,\lambda)$ designs and that all n of the designs are simple. Let $\mathcal{D} = \{D_0, \ldots, D_{n-1}\}$ be a transversal of the $t-(v,k,\lambda)$ designs. That is, any $t-(v,k,\lambda)$ design is isomorphic to precisely one D_i, $0 \leq i \leq n-1$. We wish to enumerate, for each D_i, the subsets of the set of blocks of D_i that are trades.

Suppose that $\{T_1, T_2\}$ is a trade and that $T_1 \subseteq \mathcal{B}_i$. Then, by the definition of a trade, $(D_i \setminus T_1) \cup T_2$ is also a $t-(v,k,\lambda)$ design, say D'. Now D' will be isomorphic to D_j, for some j. It may be that $i = j$, but this is not true in general. Note that, since T_1 and T_2 are non-empty and disjoint, $D' \neq D_i$. We say that T_1 is a trade *from D_i to D'* and that T_2 is a trade *from D' to D_i*.

Conversely, given D_i and any design $D' \neq D_i$, suppose that $D_i = B \cup T_1$ and $D' = B \cup T_2$, with $T_1 \cap T_2 = \emptyset$. Thus B is the set of blocks common to both designs, while $T_1, T_2 \neq \emptyset$, since $D' \neq D_i$. Now $\{T_1, T_2\}$ is a trade, and D_i contains T_1. Note that, if $D_i \cap D' = \emptyset$, B will be empty, and the trade will consist of the designs D_i and D' themselves.

So, we can generate all the trades in a design D_i by comparing D_i to every other design with the same parameters and eliminating common blocks.

Given $\mathcal{D}$, it is easy to generate all possible designs by applying all permutations of V to the designs in $\mathcal{D}$. Thus we arrive at our algorithm, given in outline below.

```
1   for all permutations of V
2       for all j in 0...n − 1
3           permute design D_j
4           for all i in 0...n − 1
5               find the trade from D_i to permuted D_j
6               store trade
7           end for
8       end for
9   end for
```

This algorithm is obviously non-polynomial, since the outer loop 1–9 will be executed $v!$ times. It is possible to enumerate all permutations so that successive permutations differ by a single transposition, see for example [25, 27]. Such an enumeration yields a *constant amortised time* (CAT) algorithm. That is, the total cost to generate all permutations, divided by the number of permutations, is a constant. So, the overhead to generate all the permutations of V is $\mathcal{O}(v!)$.

For each iteration of the loop 1–9, statement 3 will be executed n times and statements 5, 6 will be executed n^2 times each. For the designs we are concerned with, the value of v is less than 32, the word size of the computers used. Thus, each block of a design can be stored in a single word as a bit-set and set operations on the blocks performed in constant time. So the complexity of statement 3 depends only on the number of blocks in the design and will be $\mathcal{O}(b)$. For statement 5, we have to compare every block in D_i with every block in the permuted D_j and strike out matched pairs. Since the permuted design is not sorted, this will be $\mathcal{O}(b^2)$.

So, the overall complexity of the algorithm will be $\mathcal{O}(v!+v!n(b+n(b^2+S_6)))$, where S_6 stands for the complexity of statement 6. The trades are stored in binary trees, one tree to each of the n^2 ordered pairs of designs, with the trades ordered lexicographically. We call the first design (D_i) in the ordered pair the *initial* design and the second (D_j) the *final* design. For ordered pairs of designs, we regard trades as distinct if the sets of blocks of the trades in the initial designs are different, and store only the blocks of the trades in these initial designs. We do not differentiate, in the current version of the algorithm, between trades with different sets of blocks in the final designs.

Since comparing trades will be $O(b)$, searching the tree is an $O(b \log_2 NT)$ operation, where NT stands for the number of trades in the tree. Insertion of a new trade requires memory allocation and $O(b)$ time. We shall assume that the total time to execute statement 6 amortises to $O(b^2)$, or less, for each call. Thus, the complexity of the algorithm reduces to $O(v!n^2b^2)$. We shall investigate the validity of this expression empirically in Section 4. As some justification for our assumption, consider the following points, some of which are discussed further below:

i) It is possible for the trade to be empty, in which case statement 6 can be skipped.

ii) The number of distinct trades is much less than the number of times, $v!n^2$, that statement 6 is executed. So in most cases, only a partial traversal of the tree is required in statement 6 before the "new" trade is found and insertion proves unnecessary.

iii) The trees are kept as small as possible by there being one for each ordered pair of designs.

iv) The trees are initially empty, so the search time is initially small and grows as the tree grows.

Note that the running time of the algorithm is independent of the design parameters t, k, λ, except as these affect b. In practice, calculating the permutations accounts for only a minor part of the running time of the algorithm. Profiled runs of the algorithm, using the *prof* [26] Unix utility, revealed that the bulk of the time is spent building the trades in statement 5 or, to a lesser extent, comparing trades in the tree searches as part of statement 6.

As given, the algorithm takes no account of the automorphism groups of the designs. Let A_i denote the order of the automorphism group of D_i and let N_i denote the number of distinct designs isomorphic to D_i. Note that $v! = A_i N_i$, $0 \leq i \leq n-1$. Let $N = \sum_{i=0}^{n-1} v!/A_i$ be the total number of distinct designs, then the maximum number of trades from D_i is $N - 1$. The maximum number of trades from D_i to designs isomorphic to D_j, $i \neq j$, is N_j, and each of these will be enumerated at least A_j times. The maximum number of trades from D_i to isomorphic designs is $N_i - 1$ and, if the permutation is an automorphism, the trade will be empty.

Since trades to distinct designs must be different, the upper bounds given for the number of trades are always achieved, if we distinguish trades on the basis of the blocks in both T_1 and T_2. However, although trades from D_i to distinct designs are different, the sets of blocks of different trades in D_i itself may be the same. Since we store only that part of the trade which lies in D_i, the number of trades in D_i will, in general, be less than the bounds given.

Thus the algorithm may do much redundant work, generating empty trades or trades already listed. We can turn the empty trades to advantage by counting their number, for each of the initial designs. Since empty trades only arise when the permutation leaves the set of blocks of a design unchanged, this count is, in fact, the order of the automorphism group, A_i, of the design. Of course, this value is available from other sources, but its calculation here is convenient and provides a check on the operation of the algorithm.

3 Defining Sets

Given a $t-(v,k,\lambda)$ design D, a subset of the blocks of D that occurs in no other $t-(v,k,\lambda)$ design is called a *defining set* of D, and is denoted dD. A defining set, no proper subset of which is also a defining set, is called a *minimal* defining set, denoted d_mD. A defining set for which no other defining set has a smaller cardinality, is called a *smallest* defining set, denoted d_sD. We are interested here in the size, $|d_sD|$, of d_sD.

The concept of defining set was introduced in the series of articles [6, 7, 8]. The close connection between defining sets and trades is illustrated by the following two results, drawn from [9]. See also [29]. For further results on defining sets, see [10, 11, 14, 28].

Lemma 1 *Every defining set of a design $D = (V, \mathcal{B})$ contains a block of every possible trade $T_1 \subseteq \mathcal{B}$.* □

Lemma 2 *If $D = (V, \mathcal{B})$ is a $t-(v,k,\lambda)$ design and $S \subseteq \mathcal{B}$ contains a block of every minimal trade in D, then S is a defining set of D.* □

Given a collection C of k-subsets of V, any design D which contains C is said to be a *completion* of C, and C is said to *complete* to D. If C is in only one design D, and is thus a defining set, C is said to *complete uniquely*, and D is said to be the *unique completion* of C. Previous theoretical results have given lower bounds on $|d_sD|$. Consideration of the smallest volume trades in a design yields the following result.

Theorem 3 *Let $D = (V, \mathcal{B})$ be a $t-(v,k,\lambda)$ design with $|\mathcal{B}| = b$, and let s*

denote the size of a smallest volume trade in D. Then any collection, $\mathcal{B}^*$, of more than $b-s$ blocks from D completes uniquely.

Proof: If $\mathcal{B}^* = \mathcal{B}$, the result is trivial. Let $\mathcal{B}^*$ be a collection of blocks from $\mathcal{B}$, with $b-s < |\mathcal{B}^*| < b$, and suppose that $\mathcal{B}^*$ completes to two distinct designs D_1 and D_2. Now $\mathcal{B}_1 \setminus \mathcal{B}^* \neq \mathcal{B}_2 \setminus \mathcal{B}^*$, $\mathcal{B}_1 \setminus \mathcal{B}^* \neq \emptyset$ and $\mathcal{B}_2 \setminus \mathcal{B}^* \neq \emptyset$. Thus $\{\mathcal{B}_1 \setminus \mathcal{B}^*, \mathcal{B}_2 \setminus \mathcal{B}^*\}$ contains a trade with volume less than s, which is not possible. □

Corollary 4 *(i)* $|d_sD| \leq b-s+1$,
(ii) For any d_mD, $|d_mD| \leq b-s+1$.

Suppose that, given a design D, we have enumerated the family of distinct trades $\mathcal{T} = \{T_i\}_{i \in I}$ in D using our algorithm, and that $|\mathcal{T}| = d$. This family can be represented by a $d \times b$ incidence matrix $M = \{m_{ij}\}$, with $m_{ij} = 1$ if trade T_i contains block b_j and 0 otherwise. In a similar manner to [18], each row of M can be thought of as a linear inequality

$$m_{i0}b_0 + \cdots + m_{i,b-1}b_{b-1} \geq 1.$$

Here the b_i, $0 \leq i \leq b-1$, are our unknowns, and they stand for the blocks of D. They are restricted to be either 0 or 1. In view of Lemma 2, since $\mathcal{T}$ contains all trades in D, any solution to this system of d inequalities is a defining set for D. Further, in view of Lemma 1, any defining set for D will be a solution to the system of inequalities.

So we formulate the integer linear programme (ILP): minimise

$$\sum_{i=0}^{b-1} b_i,$$

subject to the system of inequalites represented by M, with $b_i = 0$ or 1, $0 \leq i \leq b-1$. Any optimal solution to this ILP yields a smallest defining set, and thus the value of $|d_sD|$. As a practical matter, to reduce the number of inequalities, we may choose to minimise $\mathcal{T}$ before solving the system, but this does not affect the validity of our argument.

To motivate what follows, consider the problem of identifying a unique design D among the N $t-(v,k,\lambda)$ designs. To do so, we need to supply information about D. If the information consists solely of blocks of D then it is a defining set. If the information also includes block intersection numbers then it is a *specifying set*, see [24]. An arbitrary collection of information about a design, sufficient to identify it uniquely, will be called an *establishing set*. If the establishing set includes information about D that is invariant under isomorphisms, then we may be able to partition $\mathcal{D}$ into two or more parts and say which part D lies in. In the limiting case, the invariants may be sufficient to identify uniquely the isomorphism class to which D belongs. Examples of design invariants include whether or not the design is simple and the order of the automorphism group of the design. An example of establishing information about D that is not an invariant is the knowledge that the design does not contain a particular k-subset of V.

Given an element D_i of $\mathcal{D}$, let D_i^* denote the set of N_i distinct designs isomorphic to D_i. For each isomorphism class D_i^* the algorithm to enumerate the trades gives the number of trades in D to designs in D_i^*. Each such set of trades represents an ILP. An optimium solution of this ILP represents the smallest number of blocks of D required to ensure that no completion lies in D_i^*, or in $D^* \setminus \{D\}$ if $D_i^* = D^*$.

If $D_i^* = D^*$, then the trades in D are all to another member of the class D^*. Such trades will be called *member trades*, or *m-trades*, and any solution, not necessarily optimal, to the ILP that the collection of such trades represents is a *member* defining set. A member defining set of D is denoted mD, and we note that, while it may have more than one completion, exactly one of these completions is in D^* and this completion is D. Just as we can define minimal and smallest defining sets, we can define minimal and smallest member defining sets. These are denoted $m_m D$ and $m_s D$ respectively.

If $D_i^* \neq D^*$, then the trades in D are all to designs in another isomorphism class. Such trades will be called *class trades*, or *c-trades*. If we form the collection of all c-trades in D then any solution, not necessarily optimal, to the ILP that the collection of such trades represents is a *class* defining set. A class defining set of D is denoted cD, and we note that, while it may have more than one completion, all of these completions are in D^* and one of them is D. Just as we can define minimal and smallest defining sets, we can define minimal and smallest class defining sets. These are denoted $c_m D$ and $c_s D$ respectively.

Note that a set of blocks that is a class defining set is a class defining set for every design to which it completes. Also, it is possible for a set of blocks to be a member defining set for more than one design, in different classes. For example, if a set of blocks S completes in only two ways, to two non-isomorphic designs D_1 and D_2, then $S = mD_1$ and $S = mD_2$.

Obviously, any defining set is also a member and a class defining set and a member and a class defining set together constitute a defining set, so

$$|m_s D|,\ |c_s D| \ \leq\ |d_s D|\ \leq\ |m_s D| + |c_s D|.$$

In the case $n = 1$, $|m_s D| = |d_s D|$ and $|c_s D| = 0$. In the case $n > 1$, for each of these inequalities, examples are given in the tables at the end of this paper where equality holds and where it does not, except that no example is known where $|d_s D| = |m_s D| + |c_s D|$. The examples in the table also show that $|m_s D| < |c_s D|$, $|m_s D| = |c_s D|$ and $|m_s D| > |c_s D|$ are all possible.

The case $|m_s D| \leq |d_s D|$ is particularly interesting, since we can find $|m_s D|$ given only the design D, by using the list of m-trades in D generated by a slightly modified version of our algorithm. We do not need a transversal of all the n classes and, in fact, need not know what n is. Since $|m_s D| = |d_s D|$ in at least two cases where $n > 1$, $|m_s D|$ is a potentially tight lower bound on $|d_s D|$. In general, $|c_s D|$ is a better lower bound for $|d_s D|$, being tight in many of the examples given, but it is not so readily calculable.

Suppose now that we wish to estimate the value of $|m_s D|$. That is, given a design $D \in D_i^*$, what is the smallest number of blocks of D that uniquely identifies it among all the designs in D_i^*? The total number of blocks in the N_i designs in D_i^* is bN_i, and the total number of k-subsets of V is $\binom{v}{k}$. So, each k-subset of V appears in an average of $bN_i/\binom{v}{k}$ designs of D_i^*. Let the factor $f = \binom{v}{k}/b$. Then $1/f$ is the average proportion of the designs in D_i^* that contain a given k-subset.

Now all designs in D_i^* are the isomorphic, differing only in the labelling of the elements. If we assume that the k-subsets of V are randomly distributed among the N_i designs of D_i^*, then f is the reciprocal of the probability that a particular k-subset of V appears in a given design. So, the knowledge that a design contains a particular k-subset of V means that the design is one of N_i/f designs from the N_i designs in D_i^*. If we assume further that the blocks in a design are independent of each other, then the knowledge of x blocks of a design means that the design is one of N_i/f^x designs from the N_i designs in D_i^*. To specify D uniquely, this value must be at most 1, that is, $N_i \leq f^x$. Taking logarithms to base f yields $x \geq \log_f N_i$. The value $\log_f N_i$ is thus the expected value of $|m_s D|$, under the assumptions stated. We will discuss the validity of this expression in Section 4.

4 Results

The algorithm was run on five sets of parameters, where $n > 1$ and all the designs are simple. The results for each of these parameter sets are described briefly in the first five subsections of this section. These results are presented in tabular form as part of the appendix at the end of this paper, with three tables per parameter set. Space precludes listing the blocks of the all designs or the trades in the designs. Instead, we content ourselves with listing the numbers of distinct trades and distinct minimal trades in each design of our transversal.

The first table (which is sometimes split into two tables, due to space limitations) lists the number of distinct trades in each design, where "distinct" means "having a different set of blocks" in the given design. The first column of this table gives the label of the design, as given in the reference from which the transversal is drawn. The next n columns list the number of distinct trades from each design to each of the other designs, that is, to designs in the given isomorphism class. The number of m-trades in D_i can be obtained from column D_i of this $n \times n$ array of values. The c column lists the total number of distinct trades in the design to non-isomorphic designs, that is, the number of c-trades. The final column lists the total number of distinct trades in the design.

Note that the number of trades listed in these last two columns can be less than the sum of the number of trades in the appropriate columns from the first n. This is due to the fact that a trade in a design can be traded in more than one way, to both isomorphic and non-isomorphic designs, and to each of these in more than one way. This last point also explains why the $n \times n$ array of values is not symmetric.

The number of distinct c-trades in D_i is bounded below by the maximum number of c-trades from D_i to each of the non-isomorphic designs. The total number of distinct trades in D_i is bounded below by the maximum of the number of m-trades and the number of c-trades. The tables contain examples where the number of c-trades and the total number of trades are equal. However, there are no examples where these numbers match those for a particular initial/final pair of designs.

There appears to be no obvious pattern to the numbers, although there seems to be a correlation with the order of the automorphism groups of the final designs, and thus the number of distinct designs in each class. Recall that the algorithm only stores the blocks of a trade in one of the designs, not both. Thus, information about how many ways a particular set of blocks can be traded is not available, although some indication of its average value can be obtained by comparing the number of distinct trades with the number of distinct designs isomorphic to the final designs, that is, the N_i.

The second table (which again is sometimes split into two tables) lists the sizes of the collections of trades in the same manner as the first, but here the collections of trades have been minimised. That is, any trade which is a proper superset of another trade in the collection has been removed. Note the significant, but very variable, reduction in the number of trades after minimisation.

Since trades in one collection may, or may not, be minimal with respect to trades in another collection, there is no obvious relationship between the number of distinct minimal c-trades, the total number of distinct minimal trades and the number of distinct minimal trades between pairs of designs. There also seems to be no relationship between the number of distinct minimal m-trades, c-trades and trades and the values of $|m_s D|$, $|c_s D|$ and $|d_s D|$.

The final table for each parameter set gives the order of the automorphism group, A_i, for each of the designs and the number of distinct designs, $N_i = v!/A_i$, in each isomorphism class. The next column gives the logarithm of this, to the base f. Recall that this logarithm gives the expected value of $|m_s D|$, under the assumption that the blocks in a design are randomly distributed and independent. This value will be discussed briefly in Subsection 4.8.

The values of $|m_s D|$, $|c_s D|$ and $|d_s D|$, found by solving the ILP optimisation problems represented by the lists of appropriate trades, are given in the final three columns. The values of $|m_s D|$ and $|c_s D|$ are all new. The values of $|d_s D|$ for four of the parameters sets have previously been calculated. Our results match the published results, except in one case, which is discussed in the relevant subsection. The values of $|d_s D|$ for the 21 $2-(10,5,4)$ designs are new.

No attempt was made to enumerate or analyse all optimal solutions, and thus all smallest defining sets, to the ILP. Two interesting questions for further investigation are whether or not a smallest defining set always contains a smallest member defining set and a smallest class defining set, and the converse question. That is, whether or not a smallest member defining set or a smallest class defining set can always be embedded in a smallest defining set.

4.1 $2-(8,4,3)$

There are four non-isomorphic $2-(8,4,3)$ designs. The transversal used here is that given in [6], which also gives $|d_s D|$ for each design.

Note that there are 30 distinct designs isomorphic to γ^* and that there are 30 distinct trades from δ^* to designs isomorphic to γ^*. So the upper bound on the number of trades given in Section 2 can be attained, for distinct trades, and all the designs isomorphic to γ^* can be generated from δ^* by trading different sets of blocks of δ^*.

4.2 $2-(10,4,2)$

There are three non-isomorphic 2–(10, 4, 2) designs, with each design being a residual design of a 2–(16, 6, 2) design. The transversal used here is that given in [12], which also gives $|d_s D|$, and enumerates all smallest defining sets, for each design.

4.3 $2-(9,4,3)$

There are 11 non-isomorphic 2–(9, 4, 3) designs. The transversal used here is that given in [23], which also gives $|d_s D|$, counts the number of distinct smallest defining sets, and lists several examples, for each design.

Note that the 11 designs can be divided into the groups $\mathcal{M}_1/\mathcal{M}_2$, $\mathcal{M}_3/\mathcal{M}_4$, $\mathcal{M}_5/\mathcal{M}_6/\mathcal{M}_7$, $\mathcal{M}_8/\mathcal{M}_9$ and $\mathcal{M}_{10}/\mathcal{M}_{11}$. Within each of these groups, the total number of distinct, or minimal distinct, trades is the same. These groupings match the possible extensions to $3-(10,5,3)$ designs, see [23]. For example, a $2-(9,4,3)$ design extends to the $3-(10,5,3)$ designs $\mathcal{N}_1$ or $\mathcal{N}_2$ if and only if it is design $\mathcal{M}_1$ or $\mathcal{M}_2$. Interestingly, $\mathcal{M}_8$ and $\mathcal{M}_9$ have more than twice as many distinct minimal c-trades and trades as any of the other designs, but they have the lowest values of both $|c_s D|$ and $|d_s D|$.

4.4 $3-(10,5,3)$

There are seven non-isomorphic $3-(10,5,3)$ designs, all of which are extensions of $2-(9,4,3)$ designs. The transversal used here is that given in [23], which also gives $|d_s D|$, counts the number of distinct smallest defining sets, and lists several examples, for each design. Note that the total number of distinct minimal c-trades in design $\mathcal{N}_4$ is less than the number of such trades from $\mathcal{N}_4$ to any of the non-isomorphic designs individually.

The value calculated for $|d_s D|$ for design $\mathcal{N}_1$ does not match the value of 6 given in [23]. The blocks of design $\mathcal{N}_1$, after sorting into lexicographic order, are:

01247, 01259, 01268, 01346, 01358, 01379, 01489, 01567, <u>02348</u>,
02357, 02369, 02456, 02789, 03459, 03678, 04578, <u>04679</u>, 05689,
12345, 12367, 12389, 12469, 12578, 13478, 13569, 14568, 14579,
<u>16789</u>, 23479, <u>23568</u>, <u>24589</u>, 24678, 25679, 34567, 34689, 35789.

The underlined values are an optimal solution of the ILP, in five blocks. That this putative defining set of five blocks from $\mathcal{N}_1$ completes uniquely was checked by performing partial completions by hand and then using the *complete* utility (see [3]) to find all completions.

First note that each pair of elements of V occurs in $\lambda_2 = 8$ blocks of the design, that each element of V occurs in $r = 18$ blocks of the design, and that the design has $b = 36$ blocks. Consider the five underlined blocks. Now, the triple 345 has not yet appeared, so the block 345xx must be in any completion three times. Each of the pairs 34, 35 and 45 has now appeared four times. Thus the blocks 34xxx, 35xxx and 45xxx must appear four more times each. The elements 3, 4 and 5 have now appeared 13, 14 and 13 times each respectively. Thus the blocks 3xxxx, 4xxxx and 5xxxx must appear 5, 4 and 5 more times each respectively. This gives, in partial form, all but two of the blocks of the design.

The element 8 has appeared four times, so must appear in fourteen other blocks. Each of the triples 348, 358 and 458 has appeared once, so each must appear twice more. We distinguish three cases, with the block 3458x occuring zero, one or two times. Taking the value of λ_2 into account, we obtain the three respective partial completions:

> 02348,04679,16789,23568,24589, 345, 345, 345, 348, 348, 34, 34, 358, 358, 35, 35, 458, 458, 45, 45, 38, 38, 3, 3, 3, 48, 48, 4, 4, 58, 58, 5, 5, 5, 8, 8;
>
> 02348,04679,16789,23568,24589, 3458, 345, 345, 348, 34, 34, 34, 358, 35, 35, 35, 458, 45, 45, 45, 38, 38, 38, 3, 3, 48, 48, 48, 4, 58, 58, 58, 5, 5, 8, -;
>
> 02348,04679,16789,23568,24589, 3458, 3458, 345, 34, 34, 34, 34, 35, 35, 35, 35, 45, 45, 45, 45, 38, 38, 38, 38, 3, 48, 48, 48, 48, 58, 58, 58, 58, 5, -, -.

These three partial completions were used as input to the *complete* utility. The first of them completed uniquely, to $\mathcal{N}_1$. The other two have no completions to $3-(10,5,3)$ designs. Thus, the set of blocks found by solving the ILP generated from the trades is a defining set, and $|d_s\mathcal{N}_1| = 5$. This being the case, the comment at the end of [23] regarding a case where the unique extension of a design has a smaller smallest defining set than the design itself does not apply.

4.5 $2-(10,5,4)$

The 21 non-isomorphic 2–(10, 5, 4) designs have been enumerated in [4, 31]. Each of these designs is embeddable in a 2–(19, 9, 4) Hadamard design, see [24, 31]. The transversal used here is that given in [31], with the bracketed numbers in the first column of Table 17 giving the numbering used in [4].

The values of $|d_sD|$ for these designs have not previously been given. For convenience, the blocks of these designs are given in Tables 18 & 19, in lexicographic order. Sample smallest member defining sets, class defining sets and defining sets are marked in these tables. The defining sets are indicated by "flagging" the blocks in them with a '·' symbol. The position of this symbol – bottom, middle, top – indicates which type of smallest defining set – member defining set, class defining set, defining set – the block is in. These defining sets were those obtained from optimal solutions to the binary ILPs. No attempt was made to find smallest member or class defining sets which were subsets of smallest defining sets, or to find smallest class defining sets which were distinct from smallest defining sets.

Note that, in most cases, $|c_sD| = |d_sD|$ and that the smallest class defining set given is also a smallest defining set. However, this is not always the case. For example, for design number five, $|c_sD| = |d_sD| = 6$ and the smallest class defining set found differs from the smallest defining set. Further, the smallest class defining set is a *proper* class defining set, in the sense that it has more than one completion. In fact it has two distinct, but isomorphic, completions. One of these is D_4, the fifth design, and the other is:

{01269, 01357, 01456, 01789, 02348, 02358, 02479, 03469, 05678, 12359, 12367, 12478, 13468, 14589, 24567, 25689, 34579, 36789}.

Note that the pairs of designs 1 & 2, 3 & 4, 6 & 7, 8 & 9, 10 & 11, 12 & 13 and 16 & 17 are complementary pairs, with the blocks of one member of the pair being the complement of the blocks of the other. Designs 20 & 21 are isomorphic to each others complement. The remaining five designs are isomorphic to their own complements. The values of $|m_sD|$, $|c_sD|$ and $|d_sD|$ are the same for these complementary pairs, as are the numbers of m-trades, c-trades and trades. Let D_i & D_{i+1} and D_j & D_{j+1} be two distinct pairs of complementary designs and suppose that D_k is one of the non-paired designs. Then we observe from the tables that the number of trades from D_i to D_{i+1} is the same as the number from D_{i+1} to D_i, the number of trades from D_k is the same to both D_i & D_{i+1}, the number of trades to D_k from both D_i & D_{i+1} is the same, and the number of trades from D_i to D_j is the same as the number from D_{i+1} to D_{j+1}, as are the number from D_i to D_{j+1} and the number from D_{i+1} to D_j.

Note that the values of $|m_sD|$ are not monotonic with the expected value $\log_{14} N_i$, with the values for designs 16 & 17 being too low for this to be the case.

4.6 Some $n = 1$ examples

Although the intended use of the algorithm is in the case where $n > 1$, it can be run where $n = 1$. We can obtain a count of the number of distinct and minimal distinct trades in these designs, all of these being m-trades. If $|d_s D|$ is not known, it can be calculated from these lists of trades. Since $|m_s D| = |d_s D|$, this provides further test data for the expression for the expected value of $|m_s D|$. Additionally, it provides a wide range of $v!n^2b^2$ values on which to perform timing tests, for complexity analysis.

Accordingly, in Table 20, we present the results of some runs in the $n = 1$ case, where the unique design is simple. The parameters of the design are listed in the first column, with the order of the automorphism group and the number of distinct designs listed in the following two columns. The next two columns contain the value of f and then the expected value of $|m_s D|$ $(= |d_s D|)$. The next two columns list the number of distinct trades and the number of distinct minimal trades respectively. The final column lists the value of $|d_s D|$, obtained by solving the ILP optimisation problems represented by the trades. These values match those available in the literature [2, 12].

Note that, for the $2-(6,3,2)$ design, the number of distinct non-minimal trades is equal to the upper bound of $N_0 - 1 = N - 1 = 11$. Thus, any $2-(6,3,2)$ design can be generated from a given design by trading a different set of blocks.

The $3-(8,4,1)$ design is an extension by complementation of the $2-(7,3,1)$ design. Hence, apart from A_0, all the values in the table are the same for both designs. The $4-(11,5,1)$ and $4-(11,6,3)$ designs are complements of each other. Hence all the values in the table are the same for both designs.

4.7 Timing Information

In an effort to validate the expression for the complexity of the algorithm, and to establish whether or not the cost of processing the trees of trades did, in fact, amortise to no more than $O(b^2)$, timing data was recorded for runs of the programme. This data is presented in Table 21, and covers all the parameter sets (for both $n > 1$ and $n = 1$) previously discussed. The first column gives the parameters of the design, with the next two columns giving n and b. The next two columns give the value of $v!n^2b^2$ and this value normalised to that for the $2-(8,4,3)$ designs. The final two columns give the running time to generate all the trades in all the designs, both actual time and normalised.

The running time is the amount of actual CPU time used by the programme, and does not include system (that is, I/O) time. These times are on a Sun-4m SPARC-based server, with a 100MHz clock. Note that the time does not include the time to process the list of trades – say to extract and minimise a particular collection of trades – nor the time to solve any ILP problem. These additional times can be significent. Times significantly less than 1 second should be interpreted with care, being close to the resolution of the Unix *time* [30] command used to obtain the running times.

The normalised running times are in good agreement with the times obtained using the expression for the complexity, with the maximum discrepancy being a factor of 2. So, tree processing times can be ignored in assessing the complexity of the algorithm and $v!n^2b^2$, suitably normalised, can be used to predict running times with high confidence.

4.8 Expected value of $|m_s D|$

To obtain an expression for the expected value of $|m_s D|$, we assumed that blocks in a design are independent. This assumption is obviously incorrect, given that the collection of blocks in a design is t-balanced. Despite this, $\log_f N_i$ turns out to be a good estimate of $|m_s D|$ for the simple designs considered here, being within 1 in all but two cases where $n > 1$ and one case where $n = 1$. The estimate is neither consistently above nor consistently below the actual value, even within a set of designs with the same parameters. Unfortunately, as already noted, $\log_f N_i$ is not monotonic with $|m_s D|$ for the $2-(10,5,4)$ designs.

Note that, the more "structure" a design has, the lower we would expect our prediction to be in relation to the actual value, since additional blocks in a defining set do not provide as much information as the initial block. As an example, where $n = 1$ and $|m_s D| = |d_s D|$, consider the $2-(11,5,2)$ design. This design is linked, with a linkage of 2, and the actual value of $|m_s D|$ is greated than the predicted value by more than 2.

5 Conclusions

The algorithm presented here, despite its simplicity, has proved effective in practice, as evidenced by the results obtained. However, due to its high complexity, extending its reach to other parameter sets will require substantial efficiency improvements, if running times are to be acceptably low.

One possible approach would be to take into account the automorphism groups of the designs. Consideration of only the distinct designs in each class would replace the $n^2v!$ term in the expression for the complexity by the term $n\sum_{i=1}^{n} v!/A_i = nN$. This represents a reduction in the amount of work by a factor of $n/(\sum_{i=1}^{n} 1/A_i)$.

This is a potentially significant reduction, with the actual value depending on the automorphism group orders. Parameter sets where one or more of the designs has automorphism group order of 1 will yield only small reductions. For the five parameter sets with $n > 1$ used, the amount of work would be reduced by factors of 26.2, 47.0, 3.0, 19.1 and 2.7 respectively. However, to realise this reduction we would have to generate a set of coset representatives of the automorphism group in the group of permutations of V. Whether or not the complexity of doing this would be outweighed by the reduction in the number of executions of statements 5 and 6 of the algorithm has not been investigated.

In cases where the running time would be too long, a modified version of the algorithm could be used to generate partial lists of trades. This could be done by imposing a limit on the time of a run or the number of trades generated, or by generating only some of the permutations. Imposing a time limit may be particularly effective, since a large number of the distinct trades are found early in the run, with the trades produced towards the end of the run being mainly duplicates of previously found trades. These partial lists of trades can be used to find lower bounds for $|m_s D|$, $|c_s D|$ and $|d_s D|$, in the manner discussed in [18], or to help eliminate sets of blocks from consideration as defining sets, as discussed in [2, 12, 13].

Currently, the complete sets of trades for all designs are dumped to data files. Separate, specially written, utilities then extract, count and minimise the required set of trades and generate the ILP optimisation problem. As the results indicate, the number of trades can be very large, and this can cause memory or disk-space problems. To overcome these it may be necessary to modify the algorithm to generate only the required set of trades when it is run, and to minimise this set of trades as it is generated.

The algorithm only stores, for each design, the blocks of the trade in the design under consideration (that is, the initial design) and does not store the full trade. This is done partly to reduce the size of the data structures needed to store the trades, and partly because, since our intended application is finding the size of defining sets, we do not need the discarded information. Similarly, distinguishing trades on the basis only of the sets of blocks in the initial design cuts down the number of "distinct" trades generated, since a set of blocks in a design may be tradable in many ways.

If the full trades are required, the algorithm could easily be modified to generate and store them.

Currently, no analysis of the set of distinct trades per se in a design has been done. Some questions raised by the results so far are:

i) In how many ways can a set of blocks be traded, and how many designs can thus be obtained from a given design by trading a given set of blocks?

ii) What proportion of (v, k, t) trades, with each t-subset occuring no more than λ times and with volume $s \leq b$, are represented in one, or all, of the $t\text{–}(v, k, \lambda)$ designs? (Note that full designs, that is, designs that consist of all k-subsets of V, obviously contain all trades.)

iii) What structure, if any, can be put on the collection of (v, k, t) trades in a design, in a class of designs, or in all designs?

iv) How does this structure relate to the module structure (see [5]) of all (v, k, t) trades?

One area that has not been addressed in this paper is that of non-simple designs. In this case, the design and the trades in a design are not sets, but multisets. The algorithm presented generates a representative of each trade which uses blocks repeated in the design, but does not generate a full list of these trades. Note that, where the trade contains blocks which are also contained in the untraded portion of the design, these duplicated blocks of the trade cannot be used to distinguish between the initial and the final designs. Given a set of parameters with $n > 1$ it is often the case that some of the designs are simple and some are non-simple. The algorithm generates a full list of trades in the simple designs, enabling $|m_s D|$, $|c_s D|$ and $|d_s D|$ to be calculated for these designs. For the non-simple designs, the partial list of trades generated could be used to find lower bounds for these values.

Acknowledgements

I would like to thank Brenton Gray and Tony Moran for fruitful discussions on trades and defining sets during the course of the work here reported. I am also grateful to Profs. Dean Hoffman and Anne Street for discussions on the algorithm during its gestational phase.

The computations were performed, using programmes developed by the author, on Sun servers and workstations in the Departments of Computer Science and of Mathematics at the University of Queensland, on the Queensland Parallel Supercomputer Facility's SP2 supercomputer, and on the Silicon Graphics' Power Challenge array supercomputer at the University of Queensland's High Performance Computing Unit. *nauty* [22] was used to check the automorphism group orders obtained from the algorithm and to cross-reference the design labellings of the $2-(10,5,4)$ designs in [4] and [31]. The utility *opbdp* [1] was used to find optimal solutions of the binary ILP optimisation problems represented by the collections of trades.

References

[1] Peter Barth, *A Davis-Putman based enumeration algorithm for linear pseudo-boolean optimisation*, Report MPI-I-95-2-003, Max-Planck-Institut für Informatik, 1995.

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

[3] Cathy Delaney, *complete* – Rationale and User's Guide, CCRR-01-95, Centre for Combinatorics, Department of Mathematics, The University of Queensland, Brisbane, Qld. 4072, Australia, 1995.

[4] Peter B. Gibbons, *Computing techniques for the construction and analysis of block designs*, Technical Report No. 92, University of Toronto, May 1976.

[5] J.E. Graver and W.B. Jurkat, *The module structure of integral designs*, Journal of Combinatorial Theory (A), **15**(1973), 75–90.

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

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

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

[9] Kenneth Raward Gray, *Special subsets of the block sets of designs*, PhD Thesis, The University of Queensland, 1990.

[10] Ken Gray and Anne Penford 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.

[11] Ken Gray and Anne Penford 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.

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

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

[14] Rebecca A.H. Gower, *Minimal defining sets in a family of Steiner triple systems*, Australasian Journal of Combinatorics, **8**(1993), 55–73.

[15] A. Hedayat and H.L. Hwang, *Construction of BIB designs with various support sizes – with special emphasis for* $v = 8$ *and* $k = 4$, Journal of Combinatorial Theory, Series A, **36**(1994), 163–173.

[16] A. Hedayat and Shuo-Yen Robert Li, *The trade-off method in the construction of BIB designs with variable support sizes*, Annals of Statistics, **7**(1979), 1277–1287.

[17] H.L. Hwang, *On the structure of* (v,k,t) *trades*, Journal of Statistical Planning and Inference, **13**(1986), 179–191.

[18] 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.

[19] G.B. Khosrovshahi, Dibyen Majumdar and Mario Widel, *On the structure of basic trades*, Journal of Combinatorics, Information & System Sciences, Vol. 17, Nos. 1–2, 102–107, (1992).

[20] Ebadollah S. Mahmoodian, *On the support size of 3-designs with repeated blocks*, Ars Combinatoria, **30**(1990), 13–22.

[21] E.S. Mahmoodian and Nasrin Soltankhah, *On the existence of* (v,k,t) *trades*, Australasian Journal of Combinatorics, **6**(1992), 279–291.

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

[23] Tony Moran, *Smallest defining sets for* $2-(9,4,3)$ *and* $3-(10,5,3)$ *designs*, Australasian Journal of Combinatorics, **10**(1994), 265–288.

[24] Tony Moran, *Defining sets for* $2-(19,9,4)$ *designs and a class of Hadamard designs*, Utilitas Mathematica, to appear.

[25] Albert Nijenhuis and Herbert S. Wilf, Combinatorial Algorithms, 2nd edition, Academic Press, 1978.

[26] *prof(1)*, Unix manual page.

[27] Frank Ruskey, Combinatorial Generation, in preparation.

[28] Anne Penford Street, *Defining sets for t-designs and critical sets for Latin squares*, New Zealand Journal of Mathematics, **21**(1992), 133–144.

[29] Anne Penford Street, *Trades and defining sets*, in Handbook of Combinatorial Designs (edited C.J. Colbourn and J.H. Dinitz), CRC Publishing Co., to appear (1995).

[30] *time(1)*, Unix manual page.

[31] J.H. van Lint, H.C.A. van Tilborg and J.R. Wiekema, *Block designs with* $v=10$, $k=5$, $\lambda=4$, Journal of Combinatorial Theory (A), **23**(1977), 105–115.

Appendix

initial design	final design					
	α^*	β^*	γ^*	δ^*	c	all
α^*	981	343	27	819	983	1508
β^*	710	251	22	535	1088	1203
γ^*	428	154	15	478	1046	1053
δ^*	1172	379	30	848	1214	1599

Table 1: The number of distinct trades in the $2-(8,4,3)$ designs.

initial design	final design					
	α^*	β^*	γ^*	δ^*	c	all
α^*	136	6	5	68	62	28
β^*	120	72	2	112	122	32
γ^*	112	56	7	16	56	56
δ^*	14	35	16	133	14	70

Table 2: The number of minimal trades in the $2-(8,4,3)$ designs.

design	A_i	N_i	$\log_5 N_i$	$\|m_s D\|$	$\|c_s D\|$	$\|d_s D\|$
α^*	12	3360	5.05	5	6	6
β^*	48	840	4.18	4	6	6
γ^*	1344	30	2.11	3	6	6
δ^*	21	1920	4.70	5	6	6

Table 3: Smallest defining set sizes of the $2-(8,4,3)$ designs.

initial design	final design H_1	H_2	H_3	c	all
H_1	66	546	1401	1461	1461
H_2	133	900	1879	1882	2049
H_3	228	1268	2102	1279	2363

Table 4: The number of distinct trades in the $2-(10,4,2)$ designs.

initial design	final design H_1	H_2	H_3	c	all
H_1	15	45	255	45	45
H_2	11	135	18	21	21
H_3	23	9	221	9	105

Table 5: The number of minimal trades in the $2-(10,4,2)$ designs.

design	A_i	N_i	$\log_{14} N_i$	$\lvert m_s D\rvert$	$\lvert c_s D\rvert$	$\lvert d_s D\rvert$
H_1	720	5040	3.23	4	8	8
H_2	48	75600	4.26	5	6	6
H_3	24	151200	4.51	5	5	5

Table 6: Smallest defining set sizes of the $2-(10,4,2)$ designs.

initial	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_{10}$	$\mathcal{M}_{11}$	c	all
$\mathcal{M}_1$	994	4423	12616	7981	17638	17134	9256	6412	3016	21076	5737	35293	35293
$\mathcal{M}_2$	1071	4506	14266	6641	21996	15262	8894	6704	3248	20722	7017	35277	35293
$\mathcal{M}_3$	1107	4967	15583	6869	21646	15163	8968	7089	3335	21563	7175	34164	35821
$\mathcal{M}_4$	1082	4912	14647	7524	19791	15894	9300	6852	3278	21686	6466	35267	35821
$\mathcal{M}_5$	1119	5065	15803	6975	21894	15354	9054	7214	3333	21392	7251	32087	35889
$\mathcal{M}_6$	1131	5053	15302	7322	21066	15774	9231	7061	3325	21481	6866	34026	35889
$\mathcal{M}_7$	1146	5026	15306	7309	20937	15693	9070	7084	3336	21447	6906	35319	35889
$\mathcal{M}_8$	980	4492	15274	6816	21449	14972	8821	7152	3234	20524	7141	33324	34228
$\mathcal{M}_9$	1000	4492	14204	7652	19532	16596	8956	7048	3268	20940	6455	33908	34228
$\mathcal{M}_{10}$	1143	5122	15857	7247	21292	15414	9099	7156	3326	21663	7083	32388	36002
$\mathcal{M}_{11}$	1081	5092	16144	6766	22036	14935	8941	7351	3361	21490	7409	35596	36002

Table 7: The number of distinct trades in the $2-(9,4,3)$ designs.

initial	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_{10}$	$\mathcal{M}_{11}$	c	all
$\mathcal{M}_1$	342	342	2592	54	3168	1080	1032	828	486	2844	828	378	378
$\mathcal{M}_2$	214	536	64	510	1136	1248	944	644	558	2676	896	378	378
$\mathcal{M}_3$	213	551	1204	770	330	933	838	627	522	1114	587	298	298
$\mathcal{M}_4$	107	663	1340	620	2046	300	604	720	390	1140	894	298	298
$\mathcal{M}_5$	218	699	742	832	1142	894	765	648	528	441	457	282	294
$\mathcal{M}_6$	148	748	1100	371	1117	885	714	696	474	343	708	286	294
$\mathcal{M}_7$	164	717	972	459	1047	885	958	687	477	339	625	294	294
$\mathcal{M}_8$	228	616	1400	824	1596	1418	1122	295	402	1592	972	978	757
$\mathcal{M}_9$	192	512	1280	624	1648	1456	1024	312	718	1632	752	808	757
$\mathcal{M}_{10}$	183	762	1025	627	417	511	646	681	506	993	564	288	298
$\mathcal{M}_{11}$	267	756	990	1134	189	1476	945	639	540	936	871	288	298

Table 8: The number of minimal trades in the $2-(9,4,3)$ designs.

design	A_i	N_i	$\log_7 N_i$	$\lvert m_s D\rvert$	$\lvert c_s D\rvert$	$\lvert d_s D\rvert$
$\mathcal{M}_1$	144	2520	4.03	4	8	8
$\mathcal{M}_2$	16	22680	5.15	5	8	8
$\mathcal{M}_3$	2	181440	6.22	5	8	8
$\mathcal{M}_4$	8	45360	5.51	5	8	8
$\mathcal{M}_5$	1	362880	6.58	6	8	8
$\mathcal{M}_6$	2	181440	6.22	5	8	8
$\mathcal{M}_7$	6	60480	5.66	5	8	8
$\mathcal{M}_8$	8	45360	5.51	5	4	6
$\mathcal{M}_9$	32	11340	4.80	5	4	6
$\mathcal{M}_{10}$	1	362880	6.58	6	8	8
$\mathcal{M}_{11}$	9	40320	5.45	5	8	8

Table 9: Smallest defining set sizes of the $2-(9,4,3)$ designs.

initial	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_3$	$\mathcal{N}_4$	$\mathcal{N}_5$	$\mathcal{N}_6$	$\mathcal{N}_7$	c	all
$\mathcal{N}_1$	3765	20151	130561	280051	9487	38296	200371	446232	448287
$\mathcal{N}_2$	4311	4723	15541	27418	7138	7456	21634	46245	46263
$\mathcal{N}_3$	4455	5291	17243	26764	8170	7792	22548	46113	48171
$\mathcal{N}_4$	4549	5393	17514	26991	8581	7915	22421	42006	48735
$\mathcal{N}_5$	3755	8856	62192	107907	7928	20752	81667	149986	156932
$\mathcal{N}_6$	3763	4724	16708	26500	7692	7940	21516	44422	45590
$\mathcal{N}_7$	4627	5458	17830	26962	8731	7945	22637	44998	49070

Table 10: The number of distinct trades in the $3-(10,5,3)$ designs.

initial	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_3$	$\mathcal{N}_4$	$\mathcal{N}_5$	$\mathcal{N}_6$	$\mathcal{N}_7$	c	all
$\mathcal{N}_1$	1619	5300	23760	34560	2556	9540	36480	24056	25271
$\mathcal{N}_2$	1664	522	54	1248	2124	630	2868	380	380
$\mathcal{N}_3$	1632	598	1362	264	2316	606	1176	306	306
$\mathcal{N}_4$	1466	749	876	1475	2352	624	357	352	370
$\mathcal{N}_5$	1184	3200	8480	12000	3326	10	10560	858	951
$\mathcal{N}_6$	1568	616	1584	1936	2234	149	1600	1562	759
$\mathcal{N}_7$	1518	807	1107	234	2358	639	1048	402	412

Table 11: The number of minimal trades in the $3-(10,5,3)$ designs.

design	A_i	N_i	$\log_7 N_i$	$\lvert m_s D\rvert$	$\lvert c_s D\rvert$	$\lvert d_s D\rvert$
$\mathcal{N}_1$	720	5040	4.38	5	4	5^*
$\mathcal{N}_2$	144	25200	5.21	5	8	8
$\mathcal{N}_3$	16	226800	6.34	6	8	8
$\mathcal{N}_4$	6	604800	6.84	6	8	8
$\mathcal{N}_5$	320	11340	4.80	5	4	6
$\mathcal{N}_6$	64	56700	5.63	5	4	6
$\mathcal{N}_7$	9	403200	6.63	6	8	8

Table 12: Smallest defining set sizes of the $3-(10,5,3)$ designs.

initial	1	2	3	4	5	6	7	8	9	10	11	12
1	18373	18487	18791	18750	13111	14098	14256	13859	14064	8553	8603	4816
2	18487	18373	18750	18791	13111	14256	14098	14064	13859	8603	8553	4762
3	18679	18685	18957	18945	13305	14313	14628	13610	13776	8458	8648	4783
4	18685	18679	18945	18957	13305	14628	14313	13776	13610	8648	8458	4721
5	18603	18603	18922	18922	12315	14267	14267	13749	13749	8614	8614	4543
6	18405	18497	19044	19320	13315	14537	13844	13852	13542	8744	8379	4623
7	18497	18405	19320	19044	13315	13844	14537	13542	13852	8379	8744	4852
8	18443	18655	17940	18156	12876	14016	13690	14137	13959	8715	8486	4758
9	18655	18443	18156	17940	12876	13690	14016	13959	14137	8486	8715	4869
10	18457	18495	17853	18526	12979	14329	13093	14467	13609	8782	8419	4642
11	18495	18457	18526	17853	12979	13093	14329	13609	14467	8419	8782	5026
12	18508	18132	17568	16684	11838	11828	13444	14424	14404	8308	8916	4974
13	18132	18508	16684	17568	11838	13444	11828	14404	14424	8916	8308	4449
14	18352	18352	17100	17100	11424	13044	13044	14492	14492	8644	8644	4710
15	16684	16684	15388	15388	11878	10576	10576	14299	14299	8275	8275	5140
16	18103	18016	16870	17496	12572	13232	12521	14449	14036	8681	8425	4825
17	18016	18103	17496	16870	12572	12521	13232	14036	14449	8425	8681	5008
18	18676	18676	19165	19165	12345	14192	14192	13940	13940	8578	8578	4521
19	18236	18236	18601	18601	13382	13891	13891	13426	13426	8479	8479	4596
20	18448	18460	19594	19900	13426	14368	14830	13309	13552	8320	8563	4759
21	18460	18448	19900	19594	13426	14830	14368	13552	13309	8563	8320	4594

Table 13: The number of distinct trades in the $2-(10,5,4)$ designs.

initial	13	14	15	16	17	18	19	20	21	c	all
1	4762	5200	2229	7242	7255	10674	8041	7514	7464	35474	36677
2	4816	5200	2229	7255	7242	10674	8041	7464	7514	35474	36677
3	4721	5148	2075	6956	7034	10729	8190	7729	7645	35724	37192
4	4783	5148	2075	7034	6956	10729	8190	7645	7729	35724	37192
5	4543	4966	2007	6965	6965	10095	7799	7825	7825	36066	37103
6	4852	5072	2007	7009	6946	10704	7950	7653	7893	36281	36964
7	4623	5072	2007	6946	7009	10704	7950	7893	7653	36281	36964
8	4869	5272	2330	7391	7325	10574	7978	7173	7260	35479	36082
9	4758	5272	2330	7325	7391	10574	7978	7260	7173	35479	36082
10	5026	5197	2303	7405	7231	10483	7841	7102	7509	36034	36241
11	4642	5197	2303	7231	7405	10483	7841	7509	7102	36034	36241
12	4449	5224	2460	7528	7636	9710	7252	6875	6363	34823	34905
13	4974	5224	2460	7636	7528	9710	7252	6363	6875	34823	34905
14	4710	5091	2416	7620	7620	10070	7308	6899	6899	35196	35253
15	5140	5026	2965	7819	7819	8602	6385	5509	5509	34069	34069
16	5008	5223	2487	7558	7624	10158	7541	6489	6912	35017	35209
17	4825	5223	2487	7624	7558	10158	7541	6912	6489	35017	35209
18	4521	5092	1966	6942	6942	10071	7656	7830	7830	36819	37263
19	4596	4996	1953	6881	6881	10442	7435	7692	7692	36533	36605
20	4594	5209	1930	6766	6859	10933	8206	7739	7834	37888	38060
21	4759	5209	1930	6859	6766	10933	8206	7834	7739	37888	38060

Table 14: The number of distinct trades in the $2-(10,5,4)$ designs, cont.

initial	1	2	3	4	5	6	7	8	9	10	11	12
1	1415	1272	924	991	1064	1317	1217	942	1006	990	775	638
2	1272	1415	991	924	1064	1217	1317	1006	942	775	990	669
3	1023	1074	1509	1460	1053	1397	960	1365	1191	959	895	676
4	1074	1023	1460	1509	1053	960	1397	1191	1365	895	959	743
5	2061	2061	1991	1991	385	1648	1648	1616	1616	1265	1265	576
6	1687	1585	1362	890	1059	1218	1379	1097	1667	845	1222	790
7	1585	1687	890	1362	1059	1379	1218	1667	1097	1222	845	585
8	739	858	1524	1340	966	1101	1489	1227	997	858	882	636
9	858	739	1340	1524	966	1489	1101	997	1227	882	858	532
10	1656	513	1371	1452	1230	777	1737	1245	1071	937	1248	750
11	513	1656	1452	1371	1230	1737	777	1071	1245	1248	937	489
12	2104	2192	1768	2448	736	1936	1208	1960	1176	1272	736	246
13	2192	2104	2448	1768	736	1208	1936	1176	1960	736	1272	279
14	1440	1440	1880	1880	840	1796	1796	1688	1688	1080	1080	170
15	2196	2196	2826	2826	1998	1638	1638	828	828	660	660	468
16	1484	1252	1832	2360	1234	1102	1586	682	684	314	1208	553
17	1252	1484	2360	1832	1234	1586	1102	684	682	1208	314	464
18	2238	2238	1853	1853	186	1503	1503	1826	1826	1212	1212	568
19	2436	2436	1592	1592	862	462	462	1560	1560	1154	1154	617
20	1449	1476	297	1782	1089	1368	954	1800	1485	1008	945	702
21	1476	1449	1782	297	1089	954	1368	1485	1800	945	1008	801

Table 15: The number of minimal trades in the $2-(10,5,4)$ designs.

initial	13	14	15	16	17	18	19	20	21	c	all
1	669	736	260	821	815	1321	1192	929	990	437	442
2	638	736	260	815	821	1321	1192	990	929	437	442
3	743	776	286	918	1032	1182	1082	778	1056	482	495
4	676	776	286	1032	918	1182	1082	1056	778	482	495
5	576	737	312	1054	1054	437	1148	1168	1168	953	463
6	585	848	280	818	982	1078	876	1054	925	512	490
7	790	848	280	982	818	1078	876	925	1054	512	490
8	532	746	230	631	693	1421	1111	1023	953	401	401
9	636	746	230	693	631	1421	1111	953	1023	401	401
10	489	723	230	342	1089	1158	1065	921	898	372	372
11	750	723	230	1089	342	1158	1065	898	921	372	372
12	279	250	224	888	520	1202	1040	1128	1252	759	513
13	246	250	224	520	888	1202	1040	1252	1128	759	513
14	170	457	256	760	760	824	1272	1140	1140	544	545
15	468	684	270	27	27	990	486	1242	1242	378	378
16	464	685	159	772	836	1246	883	1028	1186	330	330
17	553	685	159	836	772	1246	883	1186	1028	330	330
18	568	720	292	1048	1048	802	1210	1192	1192	552	497
19	617	900	278	843	843	1166	922	1264	1264	434	418
20	801	900	330	1152	1152	1224	1143	1114	1035	495	496
21	702	900	330	1152	1152	1224	1143	1035	1114	495	496

Table 16: The number of minimal trades in the $2-(10,5,4)$ designs, cont.

design	A_i	N_i	$\log_{14} N_i$	$\|m_s D\|$	$\|c_s D\|$	$\|d_s D\|$
1 (XV)	1	3628800	5.72	5	7	7
2 (XI)	1	3628800	5.72	5	7	7
3 (XII)	1	3628800	5.72	5	7	7
4 (XIV)	1	3628800	5.72	5	7	7
5 (XVI)	2	1814400	5.46	5	6	6
6 (XIX)	2	1814400	5.46	5	7	7
7 (VI)	2	1814400	5.46	5	7	7
8 (XVII)	2	1814400	5.46	5	7	7
9 (III)	2	1814400	5.46	5	7	7
10 (XX)	6	604800	5.04	5	7	7
11 (V)	6	604800	5.04	5	7	7
12 (VIII)	16	226800	4.67	5	6	7
13 (XVIII)	16	226800	4.67	5	6	7
14 (X)	16	226800	4.67	5	7	7
15 (II)	72	50400	4.10	4	8	8
16 (IV)	8	453600	4.94	4	7	7
17 (I)	8	453600	4.94	4	7	7
18 (IX)	4	907200	5.20	5	6	6
19 (VII)	8	453600	4.94	5	7	7
20 (XIII)	9	403200	4.89	5	7	7
21 (XXI)	9	403200	4.89	5	7	7

Table 17: Smallest defining set sizes of the $2-(10,5,4)$ designs.

1	2	3	4	5	6	7	8	9	10	11
01234	:01268	˙01234	˙01268	˙01234	˙01234	:01267	˙01234	:01279	:01234	:01267
01235	:01369	01235	:01369	01235	01235	01389	01235	01368	01235	:01368
01567	01459	01567	.01459	01567	.01567	˙01459	01567	˙01459	01578	01459
˙01789	˙01478	01789	01478	01789	˙01789	01468	˙01789	˙01468	˙01789	˙01469
02467	02379	02479	˙02379	˙02479	˙02478	.02379	:02478	02367	02467	02389
:02689	:02458	˙02689	02458	:02689	:02689	02458	02689	.02458	.02689	:02458
˙03489	02579	03468	˙02567	:03469	03469	:02569	.03469	˙02569	˙03479	.02579
03578	03467	:03578	03467	03578	03568	03467	:03589	:03479	:03569	03478
˙04569	.03568	.04569	:03589	:04568	04579	03578	04567	03578	:04568	03567
.12479	12378	:12467	.12378	:12478	˙12469	12368	.12469	12389	.12489	:12379
12589	12469	12589	:12469	12569	˙12589	˙12479	12568	12467	12569	˙12478
13468	˙12567	˙13489	12579	:13468	:13478	˙12578	˙13478	.12578	13468	˙12568
:13679	:13457	13679	˙13457	13679	13679	:13457	.13679	13457	˙13679	13457
14568	13589	14568	13568	14589	.14568	˙13569	˙14589	:13569	14567	13589
˙23569	˙23456	23569	23456	23589	23579	23456	23579	˙23456	23578	23456
23678	23489	.23678	23489	23678	23678	.23489	˙23678	23489	23678	23469
:24578	46789	:24578	46789	·24567	24567	46789	24579	46789	24579	46789
.34579	56789	˙34579	56789	·34579	.34589	56789	34568	56789	˙34589	56789

Table 18: The blocks, with defining sets, of the 2−(10, 5, 4) designs.

12	13	14	15	16	17	18	19	20	21
:01234	˙01267	:01234	˙01234	˙01234	:01279	:01234	.01234	:01234	:01234
:01235	:01389	˙01235	:01235	01235	˙01368	˙01235	01235	01259	˙01389
01468	01569	:01468	01467	:01468	01567	.01469	01468	˙01378	˙01478
01479	01578	˙01479	01489	01479	˙01589	·01478	.01479	˙01679	01569
02568	˙02389	02569	˙02569	02569	:02367	02568	˙02578	02457	:02358
02579	02469	.02578	˙02689	02689	02458	.02789	˙02689	.02689	02469
03678	02478	.03678	.03578	03578	˙02478	˙03589	03569	03456	.02579
03679	03456	03679	˙03789	03789	03459	03679	˙03789	˙03489	03467
04589	·03457	˙04589	.04567	.04567	03469	04567	.04567	05678	.05678
˙12689	:12367	12689	12578	˙12578	.12389	12567	12569	˙12367	12357
˙12789	.12458	12789	12678	12678	12456	˙12689	˙12678	.12468	˙12456
:13569	·12459	13569	˙13569	13569	12469	˙13579	˙13578	13589	12678
˙13578	:13468	13578	13679	13679	13457	˙13678	13679	˙14569	˙13679
14567	˙13479	14567	14589	:14589	13478	14589	.14589	.14578	14589
·23469	·23568	˙23468	˙23468	˙23467	˙23568	·23469	˙23467	23568	23689
:23478	:23579	.23479	˙23479	˙23489	:23579	:23478	˙23489	23579	24789
·24567	46789	24567	.24579	˙24579	46789	.24579	.24579	:24789	.34568
:34589	56789	˙34589	34568	.34568	56789	·34568	34568	34679	˙34579

Table 19: The blocks, with defining sets, of the 2−(10, 5, 4) designs, cont.

design	A_0	N_0	f	$\log_f N_0$	#dist	#min	$\lvert d_s D \rvert$
2−(6,3,2)	60	12	2	3.58	11	10	3
2−(7,3,1)	168	30	5	2.11	15	7	3
2−(9,3,1)	432	840	7	3.46	188	36	4
2−(11,5,2)	660	60480	42	2.95	298	66	5
3−(8,4,1)	1344	30	5	2.11	15	7	3
3−(10,4,1)	1440	2520	7	4.02	1526	415	4
4−(11,5,1)	7920	5040	7	4.38	4181	3465	5
4−(11,6,3)	7920	5040	7	4.38	4181	3465	5

Table 20: Some simple designs, with $n = 1$.

design	n	b	$v!n^2b^2$	norm	time	norm
2−(6,3,2)	1	10	72000	0.00057	0m00.03s	0.00072
2−(7,3,1)	1	7	246960	0.00195	0m00.10s	0.00239
3−(8,4,1)	1	14	7902720	0.06250	0m01.73s	0.04138
2−(9,3,1)	1	12	52254720	0.41327	0m16.42s	0.39273
2−(8,4,3)	4	14	126443520	1.00000	0m41.81s	1.00000
3−(10,4,1)	1	30	3265920000	25.8291	11m28.90s	16.4769
2−(11,5,2)	1	11	4829932800	38.1983	21m42.70s	31.1576
2−(10,4,2)	3	15	7348320000	58.1154	33m46.09s	48.4595
2−(9,4,3)	11	18	14226347520	112.511	1h44m01.72s	149.288
4−(11,5,1)	1	66	173877580800	1375.14	7h56m14.11s	683.428
4−(11,6,3)	1	66	173877580800	1375.14	8h05m24.27s	696.586
3−(10,5,3)	7	36	230443315200	1822.50	14h26m53.62s	1244.05
2−(10,5,4)	21	18	518497459200	4100.63	64h10m47.71s	5526.14

Table 21: Running times to generate all trades.