May 2002

A Search for Hadamard Matrices constructed from Williamson Matrices

J. Horton
University of Wollongong, jeffh@uow.edu.au

C. Koukouvinos
National Technical University of Athens, Greece

Jennifer Seberry
University of Wollongong, jennie@uow.edu.au

A Search for Hadamard Matrices constructed from Williamson Matrices

Abstract

We describe the implementation of a distributed computer search that uses Williamson's construction for Hadamard matrices. The search program is used to perform a complete search for matrices of orders 100 through 148. No new results are found, confirming existing results. We are convinced that no further matrices of any order less than 156 may be constructed. For reference purposes, we present tables of Hadamard matrices of orders 100 through 180 constructed using four circulant symmetric (1, -1) matrices in the Williamson array.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as Horton, J, Koukouvinos, C and Seberry, J, A Search for Hadamard Matrices constructed from Williamson Matrices, Bulletin of the Institute of Combinatorics and its Applications (ICA), 35, 2002, 75-88.

A Search for Hadamard Matrices constructed from Williamson Matrices

Jeffrey Horton[1], Christos Koukouvinos[2], and Jennifer Seberry[1]

[1] Centre for Computer Security Research,
School of Information Technology and Computer Science,
University of Wollongong,
Northfields Avenue, Wollongong, Australia
{jeffh, j.seberry}@uow.edu.au
[2] Department of Mathematics,
National Technical University of Athens,
Zografou 15773, Athens, Greece
ckoukouv@math.ntua.gr

Abstract. We describe the implementation of a distributed computer search that uses Williamson's construction for Hadamard matrices. The search program is used to perform a complete search for matrices of orders 100 through 148. No new results are found, confirming existing results. We are convinced that no further matrices of any order less than 156 may be constructed. For reference purposes, we present tables of Hadamard matrices of orders 100 through 180 constructed using four circulant symmetric $(1,-1)$ matrices in the Williamson array.

1 Introduction

An Hadamard matrix H of order n has elements ± 1 and satisfies $HH^{\mathrm{T}} = nI_n$. These matrices are used extensively in coding and communications [see Seberry and Yamada [12]]. The order of an Hadamard matrix is $n \equiv 0 \bmod 4$. The first unsolved case is order 428. We use Williamson's construction as the basis of our algorithm to construct a distributed computer search for new Hadamard matrices. We briefly describe the theory of Williamson's construction in Section 2. Previous computer searches for Hadamard matrices using Williamson's condition are described in Section 3. The implementation of the search algorithm is presented in Section 4, and the results of the search are described in Section 5.

2 Hadamard Matrices from Williamson Matrices

Theorem 1 (Williamson [16]). *Suppose there exist four* $(1,-1)$ *matrices A, B, C, D of order n which satisfy*

$$XY^{\mathrm{T}} = YX^{\mathrm{T}}, X, Y \in \{A, B, C, D\}$$

Further, suppose

$$AA^{\mathrm{T}} + BB^{\mathrm{T}} + CC^{\mathrm{T}} + DD^{\mathrm{T}} = 4nI_n \tag{1}$$

Then

$$H = \begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix} \tag{2}$$

is an Hadamard matrix of order $4n$ *constructed from a Williamson array.*

Let the matrix T be called the shift matrix. $T = (u_{ij})$, $u_{ij} = 1$ if $j - i = 1 \bmod n$ and zero elsewhere. Note that $T^n = I$, $(T^i)^{\mathrm{T}} = T^{n-i}$
Let

$$\begin{cases} A = \sum_{i=0}^{n-1} a_i T^i, & a_i = \pm 1, a_{n-i} = a_i \\ B = \sum_{i=0}^{n-1} b_i T^i, & b_i = \pm 1, b_{n-i} = b_i \\ C = \sum_{i=0}^{n-1} c_i T^i, & c_i = \pm 1, c_{n-i} = c_i \\ D = \sum_{i=0}^{n-1} d_i T^i, & d_i = \pm 1, d_{n-i} = d_i \end{cases} \tag{3}$$

Then matrices A, B, C, D may be represented as polynomials. The requirement that $x_{n-i} = x_i, x \in \{a, b, c, d\}$ forces the matrices A, B, C, D to be symmetric.

Since A, B, C, D are symmetric, (1) becomes:

$$A^2 + B^2 + C^2 + D^2 = 4nI_n$$

and the relation $XY^{\mathrm{T}} = YX^{\mathrm{T}}$ becomes $XY = YX$ which is true for polynomials.

Definition 1. *Williamson matrices are* $(1, -1)$ *symmetric circulant matrices. As a consequence of being symmetric and circulant they commute in pairs.*

We use the following theorem of Williamson's as the motivator for our search algorithm:

Theorem 2 (Williamson [16]). *If there exist solutions to the equations*

$$\mu_i = 1 + 2\sum_{j=1}^{s} t_{ij}(\omega^j + \omega^{n-j}), i = 1, 2, 3, 4 \tag{4}$$

where $s = \frac{1}{2}(n-1)$, ω *is a nth root of unity, exactly one of* $t_{1j}, t_{2j}, t_{3j}, t_{4j}$ *is nonzero and equals* ± 1 *for each* $1 \leq j \leq s$, *and*

$$\mu_1^2 + \mu_2^2 + \mu_3^2 + \mu_4^2 = 4n$$

then there exist solutions to the equations:

$$\begin{cases} A = \sum_{i=0}^{n-1} a_i T^i, & a_0 = 1, a_i = a_{n-i} = \pm 1 \\ B = \sum_{i=0}^{n-1} b_i T^i, & b_0 = 1, b_i = b_{n-i} = \pm 1 \\ C = \sum_{i=0}^{n-1} c_i T^i, & c_0 = 1, c_i = c_{n-i} = \pm 1 \\ D = \sum_{i=0}^{n-1} d_i T^i, & d_0 = 1, d_i = d_{n-i} = \pm 1 \end{cases} \tag{5}$$

That is, there exists an Hadamard matrix of order $4n$.

In matrix form, $\omega^j + \omega^{n-j}$ is represented as $T^j + T^{n-j}$. Since these are symmetric, we write

$$\omega_j = \omega^j + \omega^{n-j}$$

Remark 1. The solutions for (4) are independent of the particular root ω, so if n as defined by (1) is prime, we can choose ω so that the first μ having any ω_j assigned has ω_1. Since the equations are true for all roots of unity ω, they are also true for $\omega = 1$.

Theorem 3 (Williamson [16]). *Let n be odd, and matrices A, B, C, D satisfy (1) and (3), suppose $a_0 = b_0 = c_0 = d_0$, then exactly three of $a_j, b_j, c_j, d_j, 1 \leq j \leq n-1$, have the same sign.*

3 Results from previous searches

In many cases complete searches have been conducted for Hadamard matrices of Williamson type. Searches have also been conducted for special classes of Williamson type Hadamard matrices. Furthermore, an infinite class of such matrices is known and will also be discussed briefly.

- Baumert and Hall [2] report results of a complete search for orders $4t$, t odd and $3 \leq t \leq 23$. Some incomplete results for higher orders are also given.
- Sawade [11] reports results of a complete search for orders $4t$, $t = 25, 27$. The results for $t = 25$ were later demonstrated to be incomplete by Dokovic [4].
- Dokovic [5] reports results of a complete search for orders $4t$, $t = 29, 31$. Only a single non-equivalent solution was found for $t = 29$ and is equivalent to an earlier result due to Baumert [1].
- Koukouvinos and Kounias [10] report results of a complete search for order $4t$, $t = 33$. These results were later demonstrated to be incomplete by Dokovic [6].
- Koukouvinos and Kounias [9] report that no circulant symmetric Williamson matrices of order 39 exist, a result later demonstrated to be incorrect by Dokovic [6].
- Dokovic [6] reports results of a complete search for orders $4t$, $t = 33, 35, 39$.
- Dokovic [4] reports results of a complete search for orders $4t$, $t = 25, 37$. This extends results obtained by Sawade [11] for $t = 25$ and, for $t = 37$, by Williamson [16] and later Yamada [17] for a special class of matrices.

An infinite family of Hadamard matrices of Williamson type has been proved to exist under certain conditions [14, 15]:

Theorem 4. *If q is a prime power, $q \equiv 1 \pmod 4$, $q + 1 = 2t$, then there exists a Williamson matrix of order $4t$; we have $C = D$, and A and B differ only on the main diagonal.*

This theorem gives examples of Hadamard matrices of Williamson type for orders $4t$, $t = 31, 37, 41, 45, 49, 51, 55, \ldots$, for example.

Yamada [17] has searched for Hadamard matrices of Williamson type, with certain restrictions. These matrices are referred to as *Williamson type j matrices*. The Williamson equation for such matrices, of order $4n$ is:

$$4n = \left(1 - 2\sum_{s\in A} c_s\omega_s\right)^2 + \left(1 - 2\sum_{s\in A} c_s\omega_{sj}\right)^2 + \left(1 - 2\sum_{s\in B} d_s\omega_s\right)^2 + \left(1 - 2\sum_{s\in B} d_s\omega_{sj}\right)^2 \tag{6}$$

where $c_s, d_s = \pm 1$, $\omega_s = \omega^s + \omega^{-s}$, $\omega^n = 1$, $j^2 \equiv -1 \pmod{n}$, A, B, jA, jB is a partition of $\{1, 2, \ldots, \frac{n-1}{2}\}$. Such a j exists if and only if all prime divisors of n are $\equiv 1 \pmod 4$. This led to some new results for $n = 29, 37, 41$.

4 Search Method

4.1 Introduction

The basic search method is to examine all possible combinations of $\omega_j, 1 \leq j \leq \frac{1}{2}(n-1)$ for each $\mu_i, i = 1, 2, 3, 4$, testing each set of μ so generated to see if it satisfies Williamson's condition and can be used to form an Hadamard matrix of order $4n$. This search method is documented in more detail in the following sections.

As a result of the large size of the search space, a distributed client/server approach was taken to the problem: the server breaks work up into smaller portions which are then processed by the clients; any results discovered are reported to the server by the client. Very little work is done by the server itself.

Using a distributed approach, we are able to perform large amounts of work in a fraction of the time required for a single computer to perform the same amount of work.

At various times during the performance of the searches, Macintosh computers and computers running some variety of UNIX have been available for use. To make best use of the available resources, and to eliminate any need to install software beyond that of the client program itself, all communication was performed using low-level networking APIs, sockets [13] on UNIX and Open Transport [3] on the Macintosh, rather than using a package such as PVM [7] or MPI [8] that in some cases can facilitate the construction of distributed programs.

Searches for Hadamard matrices of all orders up to and including order 148 have been performed using Williamson's method implemented by a client/server system. Towards the end of an initial search of order 148, 37 computers were involved, 20 270MHz Ultra 5 computers from Sun Microsystems, and 17 333MHz iMacs from Apple Computer. No computers not available on the local area network were employed in the initial search. However, a subsequent search performed to verify results utilised 35 350MHz Pentium-II computers at the University of Newcastle in addition to 30 local Ultra 5 computers.

The details of the implementation of Williamson's method within the framework of a client/server system are discussed in the following sections.

4.2 Decompose $4n$ into sum-of-squares representation

The first step in performing a search is to decompose $4n$ into all possible sums-of-squares representations. Observing the form of (4), we see that when $\omega = 1$ each μ_i satisfies:

$$\begin{array}{l} |\mu_i| \equiv 1 \bmod 4, \mu_i > 0; \text{ or} \\ |\mu_i| \equiv 3 \bmod 4, \mu_i < 0. \end{array} \tag{7}$$

For example, the possible decompositions for 148 are:

$$\begin{array}{c} 1, 1, 5, 11 \\ 1, 7, 7, \ \ 7 \\ 3, 3, 3, 11 \\ 3, 3, 7, \ \ 9 \\ 5, 5, 7, \ \ 7 \end{array}$$

In the sections to follow, we write ω_{sub} to indicate some $\omega_k = \omega^k + \omega^{n-k}$ for $1 \leq k \leq \frac{1}{2}(n-1)$ when it is necessary to distinguish from an nth root of unity, ω.

4.3 Decide on the number of ω_{sub} assigned to each μ

The next step is to assign a number of ω_{sub} to each μ. Using (7), we see that if $|\mu_i| \equiv 1 \bmod 4$, then of the ω_{sub} contributing to μ_i, the number being added to μ_i will always be $\frac{|\mu_i|-1}{4}$ greater than the number of ω_{sub} that are subtracted. A similar condition can be derived for $|\mu_i| \equiv 3 \bmod 4$. These ω_{sub} are termed "fixed"; others are "floating" and always occur in pairs, one added and the other subtracted. These conditions are enforced to help limit the size of the space to be searched.

All possible permutations of the number of floating ω_{sub} are assigned to each μ over the course of the search of a particular sum-of-squares representation, subject to certain restrictions that are useful for reducing the size of the space to be searched:

1. The number of ω_{sub} assigned to μ_i must be greater than or equal to the number of ω_{sub} assigned to μ_j where $j < i$ and μ_i and μ_j correspond to the same value in the sum-of-squares decomposition. We may apply this condition because for the purposes of testing the set of μ to see if Williamson's condition is satisfied, μ_i and μ_j are interchangeable, and it is desirable to perform the test only once rather than twice. This may be extended further if more than two μ have the same value in the sum-of-squares decomposition.

2. If n is prime, then we may always place ω_1 in the first μ to which any ω_{sub} are assigned. This corresponds to solving the set of μ for some nth root of unity, ω^j, such that ω_1 is present in the first μ to which any ω_{sub} are assigned. Furthermore, if there are ω_{sub} both added and subtracted from this μ, we may either subtract or add ω_1; we do not need to check both. If this condition is in force, then condition 1 is not applied in the case of the μ to which ω_1 is assigned, but remains applicable for other μ corresponding to the same value from the sum-of-squares decomposition. Enforcing this condition can greatly reduce the size of the space to be searched: for example, applying this condition for searching for Hadamard matrices of size 148 reduces the size of the space to be searched to 37% of its size were this condition not to be enforced (reducing from about 32,387,862,644,280 to 12,062,406,963,464)[1].

For each permutation of floating ω_{sub} that is generated, we must assign specific identities to each ω_{sub} and evaluate Williamson's condition.

4.4 Assign specific identities to each ω_{sub}

We must now assign specific identities to each ω_{sub} so that Williamson's condition may be tested.

Let the number of ω_{sub} added to μ_i be represented by c_{2i-1} and the number of ω_{sub} subtracted from μ_i by c_{2i}. S_{2i-1} is the set of ω_{sub} added to μ_i and S_{2i} is the set of ω_{sub} subtracted from μ_i. That is, there are eight sets S, two for each μ. Some of these sets S may be empty.

$$\mu_i = 1 + 2 \sum_{\forall j \in S_{2i-1}} \omega_j - 2 \sum_{\forall j \in S_{2i}} \omega_j$$

Dividing ω_{sub} into two groups, one added to a μ and the other subtracted, helps to simplify the procedure for iterating over all possible combinations of ω_{sub}.

The sets S_i are formed by choosing c_i elements from the set of ω_{sub} not already allocated to an S_j, $j < i$. Recalling that $s = \frac{1}{2}(n-1)$, $S_{\text{T},0}$ is defined as:

$$S_{\text{T},0} = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_s\}.$$

$S_{\text{T},i}$ is defined as:

$$S_{\text{T},i} = S_{\text{T},i-1} - S_{i-1}, i = 1, \ldots, 8. \tag{8}$$

For convenience, we say that:

$$S_0 = \emptyset$$

[1] We would have achieved an even greater reduction in the size of the search space had we not been checking for solutions by both adding and subtracting ω_1 where this option was available. In this case, the size of space to be searched is less than half of the above figure.

Williamson's condition may be tested once $S_1, \ldots, S_8$ have been generated. All possible combinations of c_i elements from $S_{\mathrm{T},i}$ are examined; once the combinations are exhausted, the next combination for S_{i-1} is generated. The process is illustrated by the small segment of pseudocode shown in Figure 1.

```
j := 1;
do
    for k from j to 8
        populate S_{T,k} from S_{T,k-1} and S_{k-1} using (8);
        generate combination S_k by choosing c_k elements from S_{T,k};
    Test Williamson Condition using S_1,...,S_8 to generate μ_1,...,μ_4;
    j := 8;
    g := false;
    while ((j > 0) and (g == false))
        generate new combination S_j using c_j elements from S_{T,j}
        if successful
            g := true;
            j := j + 1;
        else
            j := j - 1;
while (j > 0);
```

Fig. 1. Segment of pseudocode illustrating generation of combinations for testing Williamson's condition.

So it should be easy to see that the number of tests of Williamson's condition for a particular set of $c_1, \ldots, c_8$ can be calculated as follows:

$$\text{Evaluations} = \prod_{i=1}^{8} \binom{|S_{\mathrm{T},i}|}{c_i} \tag{9}$$

Usually, however, the total number of evaluations performed will be less than this, for two reasons:

1. If condition 2 from Section 4.3 is applied, we choose one fewer ω_{sub} for the set S in which ω_1 is to appear.
2. If μ_i and $\mu_j, i < j$ correspond to the same value in the sum-of-squares decomposition of $4n$ and have the same number of ω_{sub} assigned, then we may require that if ω_x is the ω_{sub} of smallest subscript assigned to μ_i and ω_y has the smallest subscript assigned to μ_j, that $x < y$. Otherwise, work will be repeated when μ_i replicates a sequence that had previously occurred in μ_j. Enforcing this condition ensures that no repetition takes place and reduces the size of

the search space slightly. The reduction is unfortunately not as substantial as that for applying condition 2 from Section 4.3.

4.5 Dividing up the work for distribution

The obvious manner in which to reduce the amount of work performed by the clients to a reasonable level was to make the server perform part of the work described in Section 4.4. The server performs no evaluations itself, but would choose sets $S_1, \ldots, S_i$, for some $i < 8$. The client would evaluate all the possibilities for the choice of the remaining sets $S_{i+1}, \ldots, S_8$.

The server decides what value i should take by estimating the amount of work involved in a subproblem using a modification of Equation (9). Two constants $S_{\min}$ and $S_{\max}$ must be specified to the server: a subproblem is of acceptable size if its size lies between the two limits. Unfortunately, this does not yield subproblems with an even division of work: there are some very large and very small subproblems. Very small subproblems can be solved quickly, and result in a large number of reports of completed problems and requests for new problems being handled by the server over a short period of time. This can cause congestion and is not desirable.

The solution that was ultimately adopted was for the server to allocate multiple small subproblems to a client looking for work. The server also maintains a queue of pre-allocated subproblems ready for assignment to clients, so that client requests can be satisfied as rapidly as possible.

5 Search Results

Unfortunately, no new matrices were found as a result of the searches run so far. However, we are able to provide independent verification of results from previous searches. This is considered of utility since some previous searches, such as that conducted by Sawade [11], for example, failed to reveal all solutions that are now known for the order searched, in that case, order 100. In particular, we provide verification of results reported by Dokovic [6, 4] for orders 100, 140 and 148. Results for order 100 are also verified by Christos Koukouvinos.

For reference purposes, tables of Hadamard matrices derived from Williamson matrices using circulant symmetric $(1, -1)$ matrices in the Williamson array for orders 100 through 180 are presented in Appendix A. A complete search of order 156 is claimed by Dokovic [6]. Results for orders 164, 172 and 180 are incomplete.

A Tables of Hadamard Matrices of orders 100 through 180 from Williamson Matrices

Hadamard matrices of orders 100 through 180 are shown in Table 1 through Table 3 using the Williamson decomposition. In Table 4, we show matrices of order 148 using the row sums of the Williamson matrices, where each row of the solution represents the first row of one of the circulant matrices A, B, C, D.

The relationship between two current methods for classifying Williamson matrices, the Williamson decomposition of $4n$ into four squares, $s_1^2+s_2^2+s_3^2+s_4^2 = 4n$, and the row sums of the Williamson matrices m_1, m_2, m_3, m_4, is now discussed.

Lemma 1. *Let the Williamson decomposition into four squares be $s_1^2+s_2^2+s_3^2+s_4^2 = 4n$. Further, let the row sums of the four Williamson matrices A, B, C, D be m_1, m_2, m_3, m_4. Let*

$$M = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}, \underset{\sim}{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}, \underset{\sim}{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}$$

Then

$$s_1^2+s_2^2+s_3^2+s_4^2 = 4n \Leftrightarrow m_1^2+m_2^2+m_3^2+m_4^2 = 4n$$

and

$$M\underset{\sim}{s} = \underset{\sim}{m} \Leftrightarrow M\underset{\sim}{m} = \underset{\sim}{s}$$

Proof. From (4) we have, using the root $\omega = 1$, a decomposition with

$$s_i = \mu_i = 1 + 4\sum_{j=1}^{s} t_{ij},\ i = 1,2,3,4.$$

By Williamson's assumption condition,

$$s_1^2+s_2^2+s_3^2+s_4^2 = 4n.$$

On the other hand,

$$\begin{aligned} m_1 &= \sum_{j=1}^{n} a_j \\ &= 1 - 2\sum_{j=1}^{\frac{n-1}{2}} t_{1j} + 2\sum_{j=1}^{\frac{n-1}{2}} t_{2j} + 2\sum_{j=1}^{\frac{n-1}{2}} t_{3j} + 2\sum_{j=1}^{\frac{n-1}{2}} t_{4j} \\ &= 1 - \frac{1}{2}(s_1-1) + \frac{1}{2}(s_2-1) + \frac{1}{2}(s_3-1) + \frac{1}{2}(s_4-1) \\ &= \frac{1}{2}(-s_1+s_2+s_3+s_4) \end{aligned}$$

Similarly,

$$m_2 = \frac{1}{2}(s_1 - s_2 + s_3 + s_4)$$
$$m_3 = \frac{1}{2}(s_1 + s_2 - s_3 + s_4)$$
$$m_4 = \frac{1}{2}(s_1 + s_2 + s_3 - s_4)$$

and $M\underset{\sim}{s} = \underset{\sim}{m}$. Inverting we have, as $M^{-1} = M$, $M\underset{\sim}{m} = \underset{\sim}{s}$. It is easy to check that

$$m_1^2 + m_2^2 + m_3^2 + m_4^2 = s_1^2 + s_2^2 + s_3^2 + s_4^2 = 4n.$$

References

1. L. D. Baumert. Hadamard matrices of orders 116 and 232. *Bulletin of the American Mathematical Society*, 72:237, 1966.
2. L. D. Baumert and Marshall Hall, Jr. Hadamard Matrices of the Williamson Type. *Mathematics of Computation*, 19:442–447, 1965.
3. Apple Computer. Inside Macintosh: Networking with Open Transport, 1997. Available from `http://developer.apple.com/techpubs/mac/pdf/NetworkingOT.pdf`.
4. Dragomir Dokovic. Note on Williamson matrices of orders 25 and 37. *J. Combin. Math. Combin. Comput.*, 18:171–175, 1995.
5. Dragomir Z. Dokovic. Williamson matrices of order 4.29 and 4.31. *J. Combin. Theory Ser. A*, 59:442–447, 1992.
6. Dragomir Z. Dokovic. Williamson matrices of order $4n$ for n = 33, 35, 39. *Discrete Math.*, 115:267–271, 1993.
7. Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidy Sunderam. *PVM: Parallel Virtual Machine — A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, 1994. Available as Postscript from `http://www.netlib.org/pvm3/book/pvm-book.ps`.
8. William Gropp, Lush Ewing, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, 1994.
9. C. Koukouvinos and S. Kounias. There are no circulant symmetric Williamson matrices of order 39. *JCMCC*, 7:161–169, 1990.
10. Christos Koukouvinos and Stratis Kounias. Hadamard matrices of the Williamson type of order $4m$, $m = pq$: An exhaustive search for $m = 33$. *Discrete Math.*, 68:45–47, 1988.
11. K. Sawade. Hadamard matrices of order 100 and 108. *Bull. Nagoya Inst. Technology*, 29:147–153, 1977.
12. Jennifer Seberry and Mieko Yamada. Hadamard matrices, sequences and block designs. In D. J. Stinson and J. Dinitz, editors, *Contemporary Design Theory — A Collection of Surveys*, pages 431–560. John Wiley and Sons, 1992.
13. W. Richard Stevens. *UNIX Network Programming: Networking APIs: Sockets and XTI*, volume 1. Prentice Hall, second edition, 1998.

t	n	$\mu_1^2+\mu_2^2+\mu_3^2+\mu_4^2$	N?	μ_1	μ_2	μ_3	μ_4
25	100	$1^2+1^2+7^2+7^2$		1	1	$1+2\omega_6+2\omega_{12}-2\omega_2-2\omega_3-2\omega_5-2\omega_7$	$1+2\omega_8+2\omega_9-2\omega_1-2\omega_4-2\omega_{10}-2\omega_{11}$
				$1+2\omega_3-2\omega_7$	$1+2\omega_4-2\omega_1$	$1+2\omega_8-2\omega_9-2\omega_{10}-2\omega_{11}$	$1+2\omega_6-2\omega_2-2\omega_5-2\omega_{12}$
				$1+2\omega_3-2\omega_9$	$1+2\omega_4-2\omega_{12}$	$1-2\omega_1-2\omega_7$	$1+2\omega_6+2\omega_8-2\omega_2-2\omega_5-2\omega_{10}-2\omega_{11}$
25	100	$1^2+3^2+3^2+9^2$		$1+2\omega_6-2\omega_{11}$	$1+2\omega_3-2\omega_1-2\omega_{12}$	$1+2\omega_4-2\omega_7-2\omega_9$	$1+2\omega_2+2\omega_5+2\omega_{10}-2\omega_8$
				$1+2\omega_2+2\omega_{10}-2\omega_1-2\omega_8$	$1-2\omega_5$	$1+2\omega_9+2\omega_{11}-2\omega_3-2\omega_4-2\omega_7$	$1+2\omega_6+2\omega_{12}$
				1	$1+2\omega_1+2\omega_2-2\omega_3-2\omega_8-2\omega_9$	$1+2\omega_7+2\omega_{11}-2\omega_4-2\omega_6-2\omega_{12}$	$1+2\omega_5+2\omega_{10}$
25	100	$1^2+5^2+5^2+7^2$		$1+2\omega_5-2\omega_{10}$	$1+2\omega_6+2\omega_{11}-2\omega_2$	$1+2\omega_4+2\omega_9+2\omega_{12}-2\omega_7-2\omega_8$	$1-2\omega_1-2\omega_3$
25	100	$5^2+5^2+5^2+5^2$		$1+2\omega_1+2\omega_9-2\omega_6$	$1+2\omega_7+2\omega_{12}-2\omega_8$	$1+2\omega_2+2\omega_5-2\omega_4$	$1+2\omega_{10}+2\omega_{11}-2\omega_3$
				$1+2\omega_1+2\omega_9-2\omega_6$	$1+2\omega_7+2\omega_{12}-2\omega_8$	$1+2\omega_3+2\omega_5-2\omega_{11}$	$1+2\omega_4+2\omega_{10}-2\omega_2$
				$1+2\omega_1+2\omega_2-2\omega_3$	$1+2\omega_6+2\omega_9-2\omega_{10}$	$1+2\omega_7+2\omega_{11}-2\omega_4$	$1+2\omega_8+2\omega_{12}-2\omega_5$
27	108	$1^2+1^2+5^2+9^2$		1	1	$1+2\omega_6+2\omega_8+2\omega_{10}+2\omega_{13}-2\omega_1-2\omega_2-2\omega_{11}$	$1+2\omega_4+2\omega_5+2\omega_7+2\omega_{12}-2\omega_3-2\omega_9$
				$1+2\omega_1+2\omega_3-2\omega_4-2\omega_9$	$1+2\omega_2+2\omega_{12}-2\omega_{10}-2\omega_{11}$	$1+2\omega_7+2\omega_8-2\omega_6$	$1+2\omega_5+2\omega_{13}$
				$1+2\omega_9+2\omega_{13}-2\omega_5-2\omega_8$	$1+2\omega_3-2\omega_{10}$	$1+2\omega_2$	$1+2\omega_1+2\omega_6+2\omega_7+2\omega_{11}-2\omega_4-2\omega_{12}$
27	108	$1^2+3^2+7^2+7^2$		$1+2\omega_2+2\omega_5-2\omega_7-2\omega_8$	$1+2\omega_9-2\omega_{10}-2\omega_{11}$	$1+2\omega_3-2\omega_4-2\omega_6-2\omega_{13}$	$1-2\omega_1-2\omega_{12}$
				$1+2\omega_9-2\omega_4$	$1+2\omega_{11}-2\omega_5-2\omega_7$	$1+2\omega_8+2\omega_{13}-2\omega_1-2\omega_3-2\omega_6-2\omega_{10}$	$1-2\omega_2-2\omega_{12}$
27	108	$3^2+3^2+3^2+9^2$		No solutions.			
27	108	$3^2+5^2+5^2+7^2$		$1+2\omega_1-2\omega_4-2\omega_6$	$1+2\omega_{10}+2\omega_{13}-2\omega_{11}$	$1+2\omega_2+2\omega_5-2\omega_{12}$	$1+2\omega_7-2\omega_3-2\omega_8-2\omega_9$
29	116	$1^2+3^2+5^2+9^2$		$1+2\omega_2+2\omega_6+2\omega_{12}-2\omega_4-2\omega_9-2\omega_{11}$	$1+2\omega_7+2\omega_{10}-2\omega_3-2\omega_5-2\omega_8$	$1+2\omega_1$	$1+2\omega_{13}+2\omega_{14}$
29	116	$3^2+3^2+7^2+7^2$		No solutions.			
31	124	$1^2+1^2+1^2+11^2$		1	1	$1+2\omega_3+2\omega_4+2\omega_5-2\omega_6-2\omega_8-2\omega_{12}$	$1+2\omega_7+2\omega_{10}+2\omega_{15}-2\omega_1-2\omega_2-2\omega_9-2\omega_{11}-2\omega_{13}-2\omega_{14}$

Table 1. Hadamard matrices of orders 100–124 from Williamson Matrices

t	n	$\mu_1^2+\mu_2^2+\mu_3^2+\mu_4^2$	N?	μ_1	μ_2	μ_3	μ_4
31	124	$1^2+5^2+7^2+7^2$		No solutions.			
31	124	$3^2+3^2+5^2+9^2$		$1+2\omega_4-2\omega_{10}-2\omega_{15}$	$1+2\omega_{13}-2\omega_2-2\omega_{14}$	$1+2\omega_1+2\omega_3+2\omega_7-2\omega_5-2\omega_6$	$1+2\omega_8+2\omega_9+2\omega_{11}-2\omega_{12}$
31	124	$5^2+5^2+5^2+7^2$		No solutions.			
33	132	$1^2+1^2+3^2+11^2$		$1+2\omega_1+2\omega_{14}-2\omega_{13}-2\omega_{16}$	$1+2\omega_2+2\omega_5+2\omega_{11}-2\omega_6-2\omega_8-2\omega_9$	$1+2\omega_{12}-2\omega_3-2\omega_7$	$1-2\omega_4-2\omega_{10}-2\omega_{15}$
33	132	$1^2+1^2+7^2+9^2$		$1+2\omega_3+2\omega_{14}-2\omega_2-2\omega_{10}$	$1+2\omega_{11}+2\omega_{16}-2\omega_6-2\omega_8$	$1+2\omega_1-2\omega_5-2\omega_{12}-2\omega_{15}$	$1+2\omega_4+2\omega_9+2\omega_{13}-2\omega_7$
33	132	$1^2+5^2+5^2+9^2$		$1+2\omega_1+2\omega_{10}-2\omega_8-2\omega_{15}$	$1+2\omega_4+2\omega_{13}-2\omega_7$	$1+2\omega_{12}+2\omega_{14}-2\omega_2$	$1+2\omega_3+2\omega_5+2\omega_{11}+2\omega_{16}-2\omega_6-2\omega_9$
				$1+2\omega_5+2\omega_{12}-2\omega_7-2\omega_{15}$	$1+2\omega_{10}+2\omega_{16}-2\omega_2$	$1+2\omega_4+2\omega_6+2\omega_9-2\omega_1-2\omega_{13}$	$1+2\omega_3+2\omega_8+2\omega_{11}-2\omega_{14}$
33	132	$3^2+5^2+7^2+7^2$		$1+2\omega_{12}-2\omega_7-2\omega_{11}$	$1+2\omega_{14}+2\omega_{15}-2\omega_5$	$1+2\omega_2-2\omega_4-2\omega_{10}-2\omega_{16}$	$1+2\omega_1+2\omega_9-2\omega_3-2\omega_6-2\omega_8-2\omega_{13}$
35	140	$1^2+3^2+3^2+11^2$		No solutions.			
35	140	$1^2+3^2+7^2+9^2$		No solutions.			
35	140	$3^2+5^2+5^2+9^2$		No solutions.			
37	148	$1^2+1^2+5^2+11^2$		1	1	$1+2\omega_1+2\omega_3+2\omega_5+2\omega_{10}+2\omega_{17}+2\omega_{18}-2\omega_4-2\omega_9-2\omega_{12}-2\omega_{15}-2\omega_{16}$	$1+2\omega_{11}+2\omega_{14}-2\omega_2-2\omega_6-2\omega_7-2\omega_8-2\omega_{13}$
37	148	$1^2+7^2+7^2+7^2$		1	$1+2\omega_5+2\omega_7-2\omega_1-2\omega_2-2\omega_6-2\omega_{12}$	$1+2\omega_4+2\omega_{13}-2\omega_9-2\omega_{10}-2\omega_{14}-2\omega_{17}$	$1+2\omega_3+2\omega_{18}-2\omega_8-2\omega_{11}-2\omega_{15}-2\omega_{16}$
37	148	$3^2+3^2+3^2+11^2$		No solutions.			
37	148	$3^2+3^2+7^2+9^2$		No solutions.			
37	148	$5^2+5^2+7^2+7^2$		$1+2\omega_3+2\omega_4+2\omega_7-2\omega_1-2\omega_{11}$	$1+2\omega_5+2\omega_{13}+2\omega_{18}-2\omega_6-2\omega_8$	$1+2\omega_{16}-2\omega_2-2\omega_9-2\omega_{10}$	$1+2\omega_{15}-2\omega_{12}-2\omega_{14}-2\omega_{17}$
				$1+2\omega_2+2\omega_{15}+2\omega_{17}-2\omega_{13}-2\omega_{14}$	$1+2\omega_9+2\omega_{12}+2\omega_{16}-2\omega_4-2\omega_{10}$	$1-2\omega_3-2\omega_{18}$	$1+2\omega_8+2\omega_{11}-2\omega_1-2\omega_5-2\omega_6-2\omega_7$

Table 2. Hadamard matrices of orders 124–148 from Williamson Matrices (cont.)

t	n	$\mu_1^2+\mu_2^2+\mu_3^2+\mu_4^2$	N?	μ_1	μ_2	μ_3	μ_4
39	156	$1^2+3^2+5^2+11^2$		No solutions.			
39	156	$1^2+5^2+7^2+9^2$		No solutions.			
39	156	$3^2+7^2+7^2+7^2$		$1-2\omega_{13}$	$1+2\omega_5+2\omega_{17}-2\omega_3-2\omega_{10}$ $-2\omega_{11}-2\omega_{18}$	$1+2\omega_7+2\omega_{16}-2\omega_6-2\omega_8$ $-2\omega_{12}-2\omega_{14}$	$1+2\omega_1+2\omega_2-2\omega_4-2\omega_9$ $-2\omega_{15}-2\omega_{19}$
39	156	$5^2+5^2+5^2+9^2$		No solutions.			
41	164	$1^2+1^2+9^2+9^2$		1	1	$1+2\omega_1+2\omega_2+2\omega_9+2\omega_{11}$ $+2\omega_{17}+2\omega_{18}-2\omega_{12}-2\omega_{15}$ $-2\omega_{16}-2\omega_{20}$	$1+2\omega_3+2\omega_6+2\omega_8+2\omega_{10}$ $+2\omega_{13}+2\omega_{14}-2\omega_4-2\omega_5$ $-2\omega_7-2\omega_{19}$
43	172	$1^2+1^2+1^2+13^2$		$1+2\omega_1+2\omega_6+2\omega_7-2\omega_9$ $-2\omega_{11}-2\omega_{20}$	$1+2\omega_{10}+2\omega_{16}+2\omega_{17}-2\omega_3$ $-2\omega_{18}-2\omega_{21}$	$1+2\omega_5+2\omega_8+2\omega_{13}-2\omega_2$ $-2\omega_{12}-2\omega_{14}$	$1+2\omega_4+2\omega_{15}+2\omega_{19}$
45	180	$1^2+1^2+3^2+13^2$		1	1	$1+2\omega_4+2\omega_{17}+2\omega_{19}+2\omega_{21}$ $-2\omega_2-2\omega_{12}-2\omega_{13}-2\omega_{14}$ $-2\omega_{20}$	$1+2\omega_1+2\omega_6+2\omega_9+2\omega_{10}$ $+2\omega_{15}+2\omega_{16}+2\omega_{18}+2\omega_{22}$ $-2\omega_3-2\omega_5-2\omega_7-2\omega_8$ $-2\omega_{11}$

Table 3. Hadamard matrices of orders 156–180 from Williamson Matrices (cont.)

t	n	$\mu_1^2+\mu_2^2+\mu_3^2+\mu_4^2$	N?	Solution
37	148	$3^2+3^2+7^2+9^2$		`11-1-1----11--1--1111--1--11----1-1-1`
				`11-1-1----11--1--1111--1--11----1-1-1`
				`1---1----1-11-111----111-11-1----1---`
				`1111-1111-1--1---1111---1--1-1111-111`
37	148	$11^2+3^2+3^2+3^2$		`1--111-1-----1----11----1-----1-111--`
				`11111-1-----11----11----11-----1-1111`
				`1--1-1-1-11---1--1111--1---11-1-1-1--`
				`1---11-11--1-1-11----11-1-1--11-11---`
37	148	$7^2+7^2+7^2+1^2$		No solutions.
37	148	$1^2+1^2+5^2+11^2$		No solutions.
37	148	$7^2+7^2+5^2+5^2$		`11---1-----1-1-11-11-11-1-1-----1---1`
				`1--11-111------11----11------111-11--`
				`1-1111-1-11--1-1--11--1-1--11-1-1111-`
				`1--111-1----111-111111-111----1-111--`
				`1-------11-1111-1----1-1111-11-------`
				`1-1-1---1-11---1-1--1-1---11-1---1-1-`
				`1-11----11-11--11111111--11-11----11-`
				`111--111-1--1--111--111--1--1-111--11`

Table 4. Hadamard matrices of order 148 from Williamson matrices; row sums notation

14. Richard J. Turyn. An infinite class of Williamson matrices. *Journal of Combinatorial Theory Series A*, 12:319–321, 1972.
15. Albert Leon Whiteman. An infinite family of Hadamard matrices of Williamson type. *Journal of Combinatorial Theory Series A*, 14:334–340, 1973.
16. John Williamson. Hadamard's determinant theorem and the sum of four squares. *Duke Math. J.*, 11:65–81, 1944.
17. Mieko Yamada. On the Williamson type j matrices of orders 4.29, 4.41 and 4.37. *Journal of Combinatorial Theory Series A*, 27:378–381, 1979.