Symmetric Hadamard matrices of order 116 and 172 exist

Olivia Di Matteo\textsuperscript{1}, Dragomir Ž. Đoković\textsuperscript{2}, Ilias S. Kotsireas\textsuperscript{3}

Abstract

We construct new symmetric Hadamard matrices of orders 92, 116, 172. While the existence of those of order 92 was known since 1978, the orders 116 and 172 are new. Our construction is based on a recent new combinatorial array discovered by N. A. Balonin and J. Seberry. For order 116 we used an adaptation of an algorithm for parallel collision search. The adaptation pertains to the modification of some aspects of the algorithm, to make it suitable to solve a 3-way matching problem. We have also identified a new infinite series of symmetric Hadamard matrices.

1 Introduction

A Hadamard matrix of order $n$ is an $n \times n$ matrix $H = (h_{ij})$ with elements $\pm 1$ such that $HH^T = H^T H = nI_n$, where $I_n$ is the $n \times n$ identity matrix and $T$ stands for transposition. If in addition $H$ is symmetric, i.e. $h_{ij} = h_{ji}$, then it is called a symmetric Hadamard matrix. A list of currently open cases for symmetric Hadamard matrices appears in \cite{3}, page 277, Table 1.52 which states that the only 12 odd values of $v < 100$ for which a symmetric Hadamard matrix of order $4v$ is not known to exist are:

$$23, 29, 39, 43, 47, 59, 65, 67, 73, 81, 89, 93.$$  

However, symmetric conference matrices of order 46 were constructed by R. Mathon \cite{5} in 1978, and it is well known that this implies the existence of symmetric Hadamard matrices of order $2.46 = 92$. By modifying Mathon’s construction, Balonin and Seberry \cite{2} have constructed two more symmetric conference matrices of order 46 inequivalent to those of Mathon.

In this paper we construct symmetric Hadamard matrices of orders 92, 116, 172. All of them are constructed by using the GP array of Balonin and Seberry \cite{1}:

$$GP = \begin{pmatrix} A & BR & CR & DR \\ CR & D^T R & -A & -B^T R \\ BR & -A & -D^T R & C^T R \\ DR & -C^T R & B^T R & -A \end{pmatrix}, \quad (1)$$

\textsuperscript{1} University of Waterloo, Department of Physics and Astronomy, Institute for Quantum Computing e-mail: odimatte@uwaterloo.ca

\textsuperscript{2} University of Waterloo, Department of Pure Mathematics, Institute for Quantum Computing, Waterloo, Ontario, N2L 3G1, Canada e-mail: djokovic@uwaterloo.ca

\textsuperscript{3}Wilfrid Laurier University, Department of Physics & Computer Science, Waterloo, Ontario, N2L 3C5, Canada, e-mail: ikotsire@wlu.ca
where $R$ is the back-diagonal matrix obtained from the identity matrix by reversing the order of rows. In order to obtain a symmetric Hadamard matrix of order $4v$ from this array we need four circulant $\{\pm 1\}$-matrices (also known as binary matrices) $A, B, C, D$ of order $v$ such that

(i) $AA^T + BB^T + CC^T + DD^T = 4vI_v$;

(ii) $A^T = A$ and $B = C$.

Such quadruples $[A, B, C, D]$ can be constructed from suitable difference families, also known as supplementary difference sets (SDS), in the cyclic group $\mathbb{Z}_v$ consisting of four blocks. The authors of [1] refer to these quadruples as propus matrices and to the corresponding symmetric Hadamard matrix as propus-Hadamard matrix.

In the case $v = 23$, our construction is quite different from those in [3] and [2]. In the case $v = 29$ we use a method for parallel collision search. We had to adapt this method in order to be able to apply it to the problem of searching for suitable quadruples $[A, B, C, D]$. In the case $v = 43$ we in fact construct D-optimal matrices of order 86 from two binary circulants $A, D$ with $A^T = A$. The blocks $B$ and $C = B$ are provided by the Paley difference set in $\mathbb{Z}_{43}$.

The smallest order for which the existence question for symmetric Hadamard matrices is still undecided is now $4 \cdot 39 = 156$.

Moreover we give a new infinite series of symmetric Hadamard matrices derived from the series of Hadamard matrices constructed in [9, Theorem 3].

2 Some infinite series of symmetric Hadamard matrices

We summarize some results pertaining to the existence of infinite series of symmetric Hadamard matrices.

The following result is proved in [4, Corollary 4.6.5].

**Theorem 1** If $q \equiv 3 \pmod{4}$ is a prime power and $q + 2$ is a prime power, then there exists a symmetric conference matrix of order $q^2(q + 2) + 1$ and a symmetric Hadamard matrix of order $2q^2(q + 2) + 2$.

Note that for $q = 3$ one obtains a symmetric Hadamard matrix of order 92.

The following result is mentioned in [3, Theorem 1.48, p. 277].

**Theorem 2** If $n + 1$ and $n - 1$ are both odd prime powers, then there exist a symmetric regular Hadamard matrix of order $n^2$.

A list of 11 classes of orders of symmetric Hadamard matrices appears in [6], Appendix D. Some of these classes are infinite. For example class SHIII is an infinite class, as a consequence of Dirichlet’s theorem on the existence of primes in arithmetic progressions.

Another infinite series has been discovered recently in [1, Lemma 1]:

**Theorem 3** Let $q \equiv 1 \pmod{4}$ be a prime power, then propus matrices exist for orders $n = \frac{q+1}{2}$ which give propus-Hadamard matrices of order $2(q + 1)$.
This series is derived from the Turyn’s infinite series of Williamson matrices and is used to plug in into the GP array.

One of us (D.D.) has subsequently observed that the same construction is applicable to the infinite series of Hadamard matrices of Goethals–Seidel type constructed by Xia, Xia, Seberry and Wu [9, Theorem 3]:

**Theorem 4** Let \( q = 4n - 1 \) be a prime power \( \equiv 3 \) (mod 8). Then there exist an Hadamard matrix of order \( 4n \) of Goethals–Seidel type in which

\[
(I - A)^T = -I + A, \quad B^T = B \quad \text{and} \quad C = D.
\]

In fact their matrix is not just a Hadamard matrix but also a skew Hadamard matrix.

Instead of plugging in the quadruple \( [A, B, C, D] \) into the Goethals–Seidel array, we can plug the permuted quadruple \( [B, C, D = C, A] \) into the GP array to obtain a propus-Hadamard matrix. Thus we have the following theorem:

**Theorem 5** Let \( q = 4n - 1 \) be a prime power \( \equiv 3 \) (mod 8). Then there exist a symmetric Hadamard matrix of order \( 4n \) of PG-type, i.e., obtained by using the GP array.

For \( n = 11, 17, 33, 35, 53, 71, 77, 83, 123, 125 \) these symmetric Hadamard matrices are displayed on Balonin’s webpages [http://mathscinet.ru/catalogue/propus/dragomir/](http://mathscinet.ru/catalogue/propus/dragomir/)

### 3 Overview of the parallel collision search method

We give an overview of the parallel collision search in [8]. Parallel collision search is a time-memory tradeoff technique, originally designed to find collisions in the large spaces of cryptographic hash functions. It is a framework, however, which can be repurposed to suit many different problems.

For purposes of example, let \( f \) and \( g \) be hash functions; let \( D_f \) and \( D_g \) be their domains; and \( R_f \) and \( R_g \) be their ranges. We recall the definition of a collision and a claw, as follows:

**Definition 1** Two inputs, \( x, y \in D_f, x \neq y \) are said to be a collision in \( f \) if

\[
f(x) = f(y).
\]

**Definition 2** Let \( R_f = R_g \). Two inputs, \( x \in D_f, y \in D_g \) are said to be a claw of \( f \) and \( g \) if

\[
f(x) = g(y).
\]

Finding collisions is accomplished as follows. For simplicity, let us assume \( f \) has been designed such that \( D_f = R_f \). Rather than using brute force, to explore the space of \( f \) we can apply it repeatedly to an arbitrary initial input to produce a trail, or chain of values in the domain. Provided \( f \) is a roughly random hash function, trails are akin to random walks through the space. Let us designate some fraction of the elements of \( R_f \) as marked elements, called distinguished points. These are special points which have some easily testable property. If a trail reaches a point that is distinguished, it should terminate.
When beginning parallel collision finding, we set off a number of processes to generate trails, starting at random points. All processors have access to a shared memory bank, in which we store information about each trail which ends in a distinguished point. As a trail is completely deterministic, all that needs to be stored is the starting element, the final (distinguished) element, and the length of the trail. When a processor completes a trail, it attempts to store its information in the shared memory. If it sees that there is already a point in memory which ended at the same distinguished point, there are two possibilities. The first is that the trails started at the same point, in which case we are uninterested. The other possibility is that the two trails merged. The processor takes the trail it generated and the trail from storage, and traces back through their execution to find the point where they merged - if such a point is found, then we have found a collision.

This same collision finding procedure is also used to find claws. Rather than working with a single function $f$, we work with two functions $f$ and $g$. Every step in the trail requires not only an element in the domain, but also a flag, which tells the trail which function to perform in the next step. In addition to functions $f$ and $g$, we need some means of deriving a new flag for the next function, and an element appropriate to the next domain.

We now explain how we adapted the claw-finding techniques of [8] to the search for symmetric Hadamard matrices. Finding symmetric Hadamard matrices involves solving a 3-way matching problem. We are provided with three files, let’s call them $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Each file has many lines, which consist of a series of $n$ integers separated by spaces. Let us represent a line in $\mathcal{A}$ as the sequence $(a_1 \ a_2 \cdots a_n)$, and similarly for files $\mathcal{B}$ and $\mathcal{C}$. The goal of 3-way matching is to find a line in each $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ such that the element-wise sum of the three lines is equal to a pre-determined matching constant $\lambda$. In other words, we must find a triple of lines such that

$$a_i + b_i + c_i = \lambda, \quad \forall \ i \in \{1, \ldots, n\}.$$  

(4)

Let us define two functions, $f_{ab}$ and $f_c$:

$$f_{ab}(i, j) := \text{Element-wise sum of line } i \text{ from } \mathcal{A} \text{ and line } j \text{ from } \mathcal{B}$$  

(5)

$$f_c(k) := \text{Element-wise difference of } \lambda \text{ and line } k \text{ from } \mathcal{C}$$  

(6)

To execute the search, we have processors perform random walks, deterministically deciding at each step whether to execute $f_{ab}$ or $f_c$, and which lines of the file to read. To determine the next function, we concatenate the elements of the summed (or subtracted) lines, and run that through the SHA-1 hash function. A new function flag and corresponding line indices are then derived from this hash value. The definitions of $f_{ab}$ and $f_c$ are key here: for a triple of lines $(i, j, k)$ that are a solution, $f_{ab}(i, j) = f_c(k)$. This means that their concatenated values will be the same; this then implies their hash value will be the same, meaning that two trails which encounter these values will merge, and terminate at the same distinguished point (in our case, defined by properties of the binary SHA-1 hash value). If such a case occurs, then we have found a claw, and consequently a possible symmetric Hadamard matrix.

We implemented this algorithm in C++11, using Boost.MPI and this specific implementation was adapted from: [10]. Data from the files was stored in a SQLite database. As the initial files are rather large, we perform a preprocessing step before doing the matching. The
files are divided into sub-files based on the first number in each line. Then, only combinations of three files such that the first numbers sum to \( \lambda \) are actually run through the program.

As the algorithm is random and parallel, it is difficult to benchmark its runtime. We ran it on SHARCNET’s Orca cluster, a machine whose nodes have processing speeds from 2.2GHz-2.7GHz, and a minimum of 32GB RAM. We used 16 MPI processes, 14 of them generated distinguished points, one held the shared memory, and one was responsible for receiving pairs of distinguished points and checking if the trails merged and formed a claw. Searches were done for a fixed amount of time, usually 24h. Of the 15 matches found for \( v = 29 \), the shortest time taken was 238 seconds; the longest took over nine hours. All but five of the matches were found in less than three hours.

4 Results

In this section we present the construction of symmetric Hadamard matrices of orders \( 4 \cdot 23 = 92 \), \( 4 \cdot 29 = 116 \) and \( 4 \cdot 43 = 172 \). The order 92 is not new. The orders 116, 172 are new.

The solutions are listed in the form of SDSs with four base blocks. From them one can construct the corresponding binary sequences and also the circulant matrices. To be specific, we label the positions of a binary sequence of length \( v \) with \( 0, 1, \ldots, v - 1 \) in that order. To a given subset \( X \subseteq \{0, 1, \ldots, v - 1\} \) we associate the binary sequence whose \(-1\) entries occur exactly at the positions labeled by the elements of \( X \). Further, to such a binary sequence we associate the circulant matrix of order \( v \) whose first row is that sequence. These circulant matrices can be plugged in to the GP array, in a suitable order, to obtain the symmetric Hadamard matrix. We say that a block of an SDS is symmetric if the corresponding binary circulant matrix is symmetric.

4.1 Four non-equivalent solutions for \( v = 23 \)

All four SDSs have parameters \((23; 10, 10, 9, 8; 14)\) and are written as \([B, C = B, A, D]\) with \( A \) symmetric. The quadruple of corresponding circulant matrices \([A, B, C, D]\) should be plugged in to the GP array.

\[
\begin{bmatrix}
[0, 1, 2, 3, 5, 7, 9, 12, 17, 18], [0, 1, 2, 3, 5, 7, 9, 12, 17, 18],
[0, 2, 3, 6, 10, 13, 17, 20, 21], [0, 1, 2, 4, 5, 10, 13, 14],
[0, 1, 2, 3, 5, 7, 10, 11, 13, 19], [0, 1, 2, 3, 5, 7, 10, 11, 13, 19],
[0, 5, 7, 8, 11, 12, 15, 16, 18], [0, 1, 2, 7, 10, 11, 14, 16],
[0, 1, 2, 3, 6, 8, 9, 10, 14, 19], [0, 1, 2, 3, 6, 8, 9, 10, 14, 19],
[0, 1, 3, 8, 11, 12, 15, 20, 22], [0, 1, 3, 5, 7, 10, 13, 16],
[0, 1, 2, 5, 6, 8, 10, 13, 14, 16], [0, 1, 2, 5, 6, 8, 10, 13, 14, 16],
[0, 1, 3, 4, 10, 13, 19, 20, 22], [0, 1, 2, 5, 7, 12, 16, 18]
\end{bmatrix}:
\]
4.2 Fifteen non-equivalent solutions for $v = 29$

All 15 SDSs have parameters $(29; 13, 13, 11, 11; 19)$ and are written as \([A, D, B, C = B]\) with \(A\) symmetric. The quadruple of corresponding circulant matrices \([A, B, C, D]\) should be plugged in to the GP array.

\[
\begin{bmatrix}
0, 4, 5, 6, 7, 9, 13, 16, 20, 22, 23, 24, 25 \\
0, 1, 2, 3, 5, 8, 10, 13, 14, 15, 18, 22, 25 \\
0, 1, 2, 5, 6, 8, 11, 12, 14, 16, 22 \\
0, 1, 2, 5, 6, 8, 11, 12, 14, 16, 22 \\
0, 1, 5, 8, 12, 13, 14, 15, 16, 17, 21, 24, 28 \\
0, 1, 4, 5, 6, 7, 10, 11, 14, 16, 17, 19, 24 \\
0, 1, 2, 4, 7, 8, 10, 12, 15, 19, 21 \\
0, 1, 2, 4, 7, 8, 10, 12, 15, 19, 21 \\
0, 2, 6, 7, 8, 10, 11, 18, 19, 21, 22, 23, 27 \\
0, 1, 3, 4, 5, 7, 10, 13, 14, 15, 17, 20, 24 \\
0, 1, 2, 5, 6, 7, 9, 12, 14, 20, 23 \\
0, 1, 2, 5, 6, 7, 9, 12, 14, 20, 23 \\
0, 2, 3, 4, 6, 11, 13, 16, 18, 23, 25, 26, 27 \\
0, 1, 2, 5, 6, 8, 10, 11, 14, 17, 18, 20, 22 \\
0, 1, 2, 3, 6, 7, 11, 13, 14, 17, 22 \\
0, 1, 2, 3, 6, 7, 11, 13, 14, 17, 22 \\
0, 6, 8, 9, 10, 12, 13, 16, 17, 19, 20, 21, 23 \\
0, 1, 3, 5, 6, 7, 10, 12, 13, 16, 18, 21, 22 \\
0, 1, 2, 3, 7, 9, 12, 14, 17, 18, 22 \\
0, 1, 2, 3, 7, 9, 12, 14, 17, 18, 22 \\
0, 1, 5, 8, 12, 13, 14, 15, 16, 17, 21, 24, 28 \\
0, 1, 2, 3, 6, 8, 9, 12, 13, 15, 19, 20, 23 \\
0, 1, 2, 5, 7, 9, 11, 12, 15, 17, 20 \\
0, 1, 2, 5, 7, 9, 11, 12, 15, 17, 20 \\
0, 1, 2, 3, 6, 8, 9, 11, 12, 16, 18, 20, 25 \\
0, 1, 2, 3, 6, 8, 9, 11, 12, 16, 18, 20, 25 \\
0, 5, 6, 8, 9, 13, 16, 20, 21, 23, 24 \\
0, 1, 2, 3, 4, 7, 8, 11, 13, 17, 19 \\
0, 1, 2, 3, 4, 6, 9, 10, 11, 14, 17, 19, 23 \\
0, 1, 2, 3, 4, 6, 9, 10, 11, 14, 17, 19, 23 \\
0, 2, 7, 11, 12, 14, 15, 17, 18, 22, 27 \\
0, 1, 2, 3, 6, 9, 13, 14, 18, 20, 24 \\
0, 1, 2, 3, 5, 6, 9, 11, 12, 15, 17, 22, 24 \\
0, 1, 2, 3, 5, 6, 9, 11, 12, 15, 17, 22, 24 \\
0, 2, 6, 7, 8, 11, 18, 21, 22, 23, 27 \\
0, 1, 2, 3, 5, 6, 10, 13, 14, 17, 21 \\
0, 1, 2, 4, 5, 6, 9, 10, 13, 15, 16, 17, 23 \\
0, 1, 2, 4, 5, 6, 9, 10, 13, 15, 16, 17, 23 \\n\end{bmatrix}
\]
4.3 One solution for $v = 43$

In this case we start by constructing D-optimal matrices of order 86 by using the well known two-circulant construction. The parameter set of the relevant SDS, $[D, A]$, is $(43; 21, 15; 15)$. The important feature of this SDS is that one of the sets has to be symmetric. Our example is the following:

\[
\begin{bmatrix}
1, 2, 7, 10, 12, 17, 19, 22, 27, 28, \\
0, 1, 3, 5, 7, 10, 13, 16, 19, 21, 25, \\
0, 1, 3, 5, 7, 8, 10, 12, 13, 14, 18, 21, 22, \\
0, 1, 3, 5, 7, 8, 10, 12, 13, 14, 18, 21, 22, \\
0, 3, 6, 11, 12, 13, 16, 17, 18, 23, 26, \\
0, 1, 2, 3, 4, 6, 10, 12, 16, 17, 20, \\
0, 1, 2, 3, 5, 7, 8, 10, 14, 16, 19, 20, 24, \\
0, 1, 2, 3, 5, 7, 8, 10, 14, 16, 19, 20, 24, \\
0, 1, 2, 4, 10, 11, 18, 19, 25, 27, 28, \\
0, 1, 4, 5, 7, 9, 12, 13, 16, 20, 23, \\
0, 1, 2, 4, 6, 7, 8, 10, 11, 14, 17, 19, 22, \\
0, 1, 2, 4, 6, 7, 8, 10, 11, 14, 17, 19, 22, \\
0, 1, 2, 8, 10, 14, 15, 19, 21, 27, 28, \\
0, 1, 2, 5, 7, 10, 14, 15, 18, 19, 24, \\
0, 1, 3, 4, 6, 7, 8, 11, 13, 15, 17, 22, 23, \\
0, 1, 3, 4, 6, 7, 8, 11, 13, 15, 17, 22, 23, \\
0, 1, 9, 11, 12, 14, 15, 17, 18, 20, 28, \\
0, 1, 2, 5, 6, 10, 12, 17, 18, 21, 26, \\
0, 1, 2, 3, 5, 7, 9, 11, 12, 14, 15, 20, 24, \\
0, 1, 2, 3, 5, 7, 9, 11, 12, 14, 15, 20, 24, \\
0, 5, 6, 8, 9, 13, 16, 20, 21, 23, 24, \\
0, 1, 4, 5, 7, 8, 10, 16, 17, 18, 23, 24
\end{bmatrix}:
\]

5 Acknowledgements

The authors wish to acknowledge generous support by NSERC. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET) and Compute/Calcul Canada. We thank N. A. Balonin and J. Seberry for sending us copies of their papers [1, 2] and Michele Mosca for useful discussions. O.D.M. would like to thank
Gary Graham for some suggestions regarding the adaptation of the parallel collision finding implementation.

References

[1] N. A. Balonin, Jennifer Seberry, Visualizing Hadamard matrices: the propus construction, preprint 15pp (submitted 6 Aug 2014).

[2] N. A. Balonin, Jennifer Seberry, A review and new symmetric conference matrices, Informatsionno-upravliaiushchie sistemy, 2014, 8470; 4 (71), 2–7.

[3] R. Craigen and H. Kharaghani, Hadamard Matrices and Hadamard Designs. In Handbook of Combinatorial Designs. Edited by Charles J. Colbourn and Jeffrey H. Dinitz. Second edition. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2007.

[4] Y. J. Ionin and M. S. Shrikhande, Combinatorics of Symmetric Designs. New Mathematical Monographs, 5. Cambridge University Press, Cambridge, 2006.

[5] R. Mathon, Symmetric conference matrices of order $pq^2 + 1$. Canad. J. Math., 30 (1978), 321–331.

[6] J. Seberry Wallis, Hadamard Matrices, in W. D. Wallis, A. Penfold Street, Jennifer Seberry Wallis, Combinatorics: Room squares, sum-free sets, Hadamard matrices. Lecture Notes in Mathematics, Vol. 292. Springer-Verlag, Berlin-New York, 1972.

[7] R. J. Turyn, An infinite class of Williamson matrices, J. Combinatorial Theory Ser. A 12 (1972), 319–321.

[8] Paul C. van Oorschot and Michael J. Wiener, Parallel collision search with cryptanalytic applications, Journal of Cryptology, January 1999, Volume 12, Issue 1, 1–28.

[9] M. Xia, T. Xia, J. Seberry and J. Wu, An infinite series of Goethals-Seidel arrays, Discrete Applied Mathematics 145 (2005), 498–504.

[10] O. Di Matteo, Parallelizing quantum circuit synthesis (Currently unpublished MSc thesis), University of Waterloo, 2015.