A Heuristic Procedure with Guided Reproduction for Constructing Cocyclic Hadamard Matrices

V. Álvarez, M.D. Frau, and A. Osuna*

Dpto. Matemática Aplicada I, Universidad de Sevilla, Avda. Reina Mercedes s/n
41012 Sevilla, Spain
{valvarez,mdfrau,aosuna}@us.es

Abstract. A genetic algorithm for constructing cocyclic Hadamard matrices over a given group is described. The novelty of this algorithm is the guided heuristic procedure for reproduction, instead of the classical crossover and mutation operators. We include some runs of the algorithm for dihedral groups, which are known to give rise to a large amount of cocyclic Hadamard matrices.

1 Introduction

A Hadamard matrix is a $n \times n$ square $(-1, 1)$ matrix H_n so that $H_n \cdot H_n^T = nI$. Equivalently, a Hadamard matrix is a square matrix over $\{1, -1\}$ so that its rows are pairwise orthogonal.

The knowledge of Hadamard matrices is a major question for applications in a wide range of different disciplines, as in the design of good (even optimal) error-correcting codes meeting the Plotkin bounds (see [15] for details). A classical reference on Hadamard matrices and their uses is [9].

It may be easily proved that the size n of a Hadamard matrix H_n must be 1, 2 or a multiple of 4. It is conjectured that such a H_n exists for all n divisible by 4. However, the proof of this conjecture remains an important problem in Coding Theory, since there is no evidence of this fact until now.

In fact, there are infinitely many orders multiple of four for which uncertainty about the existence of these matrices has not been removed at all. Furthermore, even in the case that a Hadamard matrix is known to exist for a given order $n = 4t$, there is no algorithm available which outputs a Hadamard matrix of this order $4t$ in reasonable time, as it is pointed out in [14].

The *cocyclic* framework concerning Hadamard matrices was introduced in the 90s [12,13] as a promising context to solve the questions above.

A cocyclic matrix M_f over a finite group $G = \{g_1, \ldots, g_{4t}\}$ of order $|G| = 4t$ consists in a matrix $M = (f(g_i, g_j))$, $f : G \times G \to \{1, -1\}$ being a 2-cocycle over G with coefficients in $\{1, -1\}$, so that

$$f(g_i, g_j)f(g_ig_j, g_k) = f(g_j, g_k)f(g_i, g_jg_k), \qquad \forall\, g_i, g_j, g_k \in G$$

* All authors are partially supported by the research projects FQM–296 and P07–FQM–02980 from JJAA and MTM2008-06578 from MICINN (Spain).

The link between cocyclic and Hadamard matrices was first noticed in [12]. A more recent reference is [11], in which many of the classical and more recently discovered constructions of Hadamard matrices are shown to be cocyclic. This support the idea that cocyclic construction is the most uniform construction technique for Hadamard matrices yet known. Consequently, the *cocyclic Hadamard Conjecture* arises in turn.

The main advantages of working with cocyclic Hadamard matrices may be resumed in the following facts:

- The cocyclic Hadamard test (which claims that it suffices to check whether the summation of every row but the first is zero, see [13] for details) runs in $O(t^2)$ time, better than the $O(t^3)$ algorithm for usual (not necessarily cocyclic) Hadamard matrices.
- The search space is reduced to the set of cocyclic matrices over a given group (that is, 2^s matrices, provided that a basis for 2-cocycles over G consists of s generators), instead of the whole set of $\binom{\binom{4t}{2t}}{4t-1}$ matrices with entries in $\{-1,1\}$ consisting of the row $(1,\overset{t}{\ldots},1)$ and $4t-3$ vectors of length $4t$ orthogonal to $(1,\overset{t}{\ldots},1)$.

In particular, the work in [5] suggest that the cocyclic framework (c.f. in the table below) may reduce significantly the size of the search space in the general framework (g.f. for brevity) case, as the table below indicates:

t	1	2	3	4	5	6	7	8
c.f.	$O(10^0)$	$O(10^1)$	$O(10^2)$	$O(10^3)$	$O(10^5)$	$O(10^6)$	$O(10^7)$	$O(10^8)$
g.f.	$O(10^1)$	$O(10^9)$	$O(10^{24})$	$O(10^{49})$	$O(10^{82})$	$O(10^{125})$	$O(10^{177})$	$O(10^{238})$

Considerable effort has been devoted to the design of efficient algorithms for constructing cocyclic Hadamard matrices. Exhaustive search is not feasible for orders $4t$ greater than 20 (the search space grows exponentially on t, see [5] for instance). Consequently, alternative methods are required. As far as we know, two different heuristic methods have been proposed until now, in terms of image restorations [6] and genetic algorithms [2].

We present here a new genetic algorithm for constructing cocyclic Hadamard matrices. The main difference with respect to that of [2] is a novel heuristic for reproduction: instead of the usual crossover and mutation operators we shall better use a guided reproduction procedure. Calculations in Section 5 suggest that this new feature improves the original algorithm. This heuristic involves the notions of *i-paths* and *intersections* introduced in [5], to be described further in Section 2.

As it is shown in [5], dihedral groups seems to be the most prolific familiy of groups giving rise to cocyclic Hadamard matrices. We particularize the algorithm to the case of these groups. We also include some runs of the algorithm, which have been worked out in Mathematica 4.0, running on a *Pentium IV 2.400 Mhz DIMM DDR266 512 MB*.

A deeper study on the way in which 2-coboundaries over G have to be combined in order to give rise to cocyclic Hadamard matrices (attending to i-paths and intersections, as described in [5]) would lead to an improvement of the performance of the guided genetic algorithm in a straightforward manner.

We organize the paper as follows. Section 2 collects some general notions and results about cocyclic Hadamard matrices. The algorithm looking for cocyclic Hadamard matrices equipped with the new heuristic for reproduction is described in Section 3. Section 4 is devoted to particularize the algorithm to the case of dihedral group.

2 Generalities about Cocyclic Hadamard Matrices

Consider a multiplicative group $G=\{g_1 = 1, g_2, \ldots, g_{4t}\}$, not necessarily abelian. A cocyclic matrix M_f over G consists in a binary matrix $M_f = (f(g_i, g_j))$ coming from a 2-cocycle f over G, that is, a map $f : G \times G \to \{1, -1\}$ such that

$$f(g_i, g_j)f(g_ig_j, g_k) = f(g_j, g_k)f(g_i, g_jg_k), \qquad \forall\, g_i, g_j, g_k \in G.$$

We will only use normalized cocycles f (and hence normalized cocyclic matrices M_f), so that $f(1, g_j) = f(g_i, 1) = 1$ for all $g_i, g_j \in G$ (and correspondingly $M_f = (f(g_i, g_j))$ consists of a first row and column all of 1s).

Effective methods for constructing a basis $\mathcal{B}$ for 2-cocycles over a given group G are known ([12,13],[7],[4]). Such a basis consists of some representative 2-cocycles (coming from inflation and transgression) and some elementary 2-coboundaries ∂_i, so that every cocyclic matrix admits a unique representation as a Hadamard (pointwise) product $M = M_{\partial_{i_1}} \ldots M_{\partial_{i_w}} \cdot R$, in terms of some coboundary matrices $M_{\partial_{i_j}}$ and a matrix R formed from representative cocycles.

Recall that every *elementary coboundary* ∂_d is constructed from the characteristic set map $\delta_d : G \to \{\pm 1\}$ associated to an element $g_d \in G$, so that

$$\partial_d(g_i, g_j) = \delta_d(g_i)\delta_d(g_j)\delta_d(g_ig_j) \qquad \text{for} \qquad \delta_d(g_i) = \begin{cases} -1 & g_d = g_i \\ 1 & g_d \neq g_i \end{cases} \tag{1}$$

Although the elementary coboundaries generate the set of all coboundaries, they might not be linearly independent (see [4] for instance). Moreover, since the elementary coboundary ∂_{g_1} related to the identity element in G is not normalized, we may assume that $\partial_{g_1} \notin \mathcal{B}$.

The cocyclic Hadamard test asserts that a cocyclic matrix is Hadamard if and only if the summation of each row (but the first) is zero [13]. In what follows, the rows whose summation is zero are termed *Hadamard rows*.

We now reproduce the notions of *generalized coboundary matrix*, *i-walk* and *intersection* introduced in Definition 2 of [5].

The *generalized coboundary matrix* $\bar{M}_{\partial_j}$ related to a elementary coboundary ∂_j consists in negating the j^{th}-row of the matrix M_{∂_j}. Note that negating a row of a matrix does not change its Hadamard character. As it is pointed out in [5], every generalized coboundary matrix $\bar{M}_{\partial_j}$ contains exactly two negative entries in each row $s \neq 1$, which are located at positions (s,i) and (s,e), for $g_e = g_s^{-1} g_i$. We will work with generalized coboundary matrices from now on.

A set $\{\bar{M}_{\partial_{i_j}} : 1 \leq j \leq w\}$ of generalized coboundary matrices defines an *i-walk* if these matrices may be ordered in a sequence $(\bar{M}_{l_1}, \ldots, \bar{M}_{l_w})$ so that consecutive matrices share exactly one negative entry at the i^{th}-row. Such a walk is called an *i-path* if the initial and final matrices do not share a common -1, and an *i-cycle* otherwise. As it is pointed out in [5], every set of generalized coboundary matrices may be uniquely partitioned into disjoint maximal i-walks.

A characterization of Hadamard rows may be easily described attending to i-paths.

Proposition 1. *[5] The i^{th} row of a cocyclic matrix $M = M_{\partial_{i_1}} \ldots M_{\partial_{i_w}} \cdot R$ is a Hadamard row if and only if*

$$2c_i - 2I_i = 2t - r_i \tag{2}$$

where c_i denotes the number of maximal i-paths in $\{\bar{M}_{\partial_{i_1}}, \ldots, \bar{M}_{\partial_{i_w}}\}$, r_i counts the number of -1s in the i^{th}-row of R and I_i indicates the number of positions in which R and $\bar{M}_{\partial_{i_1}} \ldots \bar{M}_{\partial_{i_w}}$ share a common -1 in their i^{th}-row.

From now on, we will refer to the positions in which R and $\bar{M}_{\partial_{i_1}} \ldots \bar{M}_{\partial_{i_w}}$ share a common -1 in a given row simply as *intersections*, for brevity.

Equation (2) is the heart of the guided heuristic procedure for reproduction which is applied in the genetic algorithm described in this paper.

3 The Algorithm

The genetic algorithm described in [2] and implemented in [3] is based upon the natural evolution principles of Holland's [10]:

- The population consists of a subset of $4t$ cocyclic matrices M_f over G, $M_f = (f(g_i, g_j))$, which are identified to a binary tuple, the coordinates $(f_1, \ldots, f_s)$ of the 2-cocycle f with regards to the basis $\mathcal{B}$. Accordingly, the coordinates f_i are the genes of the individual f.
- The evaluation function counts the number of Hadamard rows in M_f: the more Hadamard rows M_f posses, the fittest M_f is. In particular, an individual *ind* gives rise to a cocyclic Hadamard matrix if and only if evaluate(*ind*)$= 4t - 1$.
- Crossover combines the features of two parent chromosomes to form two similar offspring by swapping corresponding segments of the parents.
- Mutation arbitrarily alters just one gene of a selected individual (the mutation rate is fixed in 1%).

In the reproduction process, the individuals of the population are paired at random, so that the application of the crossover operator gives rise to another $4t$ individuals, which are added to the population. The generation $i+1$ is formed from generation i by choosing the $4t$ fittest individuals after the reproduction process.

We now propose a different approach. Instead of the usual crossover and mutation operators described above, we shall better use another heuristic for reproduction. With probability p_r[1], an individual M_f randomly selected from the population gives rise to $4t-1$ children, so that the $(i+1)^{th}$-row of the i^{th}-child is Hadamard. Otherwise the usual crossover operator is used, applied over two individuals randomly selected. Generation P_{w+1} is obtained from generation P_w keeping the fittest individuals and replacing a set of less fit individuals with the children just constructed, so that a population of $8t$ individuals is formed. In this process duplicate copies of the same individual are not permitted.

Consequently, the blinded processes of crossover and mutation are now substituted by a completely oriented procedure for reproduction: this way it is guaranteed that anytime an individual exists such that its i^{th}-row is Hadamard.

In order to generate these children, the genes of M_f have to be modified so that equation (2) is satisfied. It is remarkable that the magnitudes c_i and I_i depends heavily on the subset of 2-coboundaries which gives rise to M_f. On the contrary, the magnitude r_i depends only on the representative 2-cocycles implicated in the generation of M_f.

Attending to these facts, a heuristic procedure for reproduction may be straightforwardly defined in the following way. The key idea is to modify the genes of M_f corresponding to 2-coboundaries in such a manner that the magnitudes c_i and I_i are also modified in turn, so that the difference $2c_i - 2I_i$ is closer to the constant value $2t - r_i$.

Depending on whether $2c_i - 2I_i > 2t - r_i$ or $2c_i - 2I_i < 2t - r_i$, we need to increase or decrease I_i (resp. decrease or increase c_i) so that the equality may hold. More concretely:

1. If $2c_i - 2I_i > 2t - r_i$, the algorithm randomly chooses one of the following possibilities:
   - Collapses two different i-paths into just one i-path, so that c_i decreases 1 unit.
   - Introduces a new negative sharing position between R and the product of M_{∂_j}, so that I_i increases 1 unit.
2. If $2c_i - 2I_i < 2t - r_i$, the algorithm randomly chooses one of the following possibilities:
   - Splits one i-path into two different i-paths, so that c_i increases 1 unit.
   - Adds a new i-path, introducing a new 2-coboundary generator, so that c_i increases 1 unit.
   - Eliminates a negative sharing position between R and the product of M_{∂_j}, so that I_i decreases 1 unit.

The way in which these procedures have to be implemented depends on the group G over which 2-cocycles are considered. In the following section we will

[1] Experimental results show that a good value for the parameter p_r is 0.8.

explicitly show a pseudo-code of the particular heuristic procedure for reproduction in the case of dihedral groups.

The population is expected to evolve generation through generation until an optimum individual (i.e. a cocyclic Hadamard matrix) is located. This has been the case in the examples showed in the last section.

We include now a pseudo-code of the algorithm.

```
Input: a group (G,·) of order |G| = 4t
Output: some (eventually one) cocyclic Hadamard matrices over G

\\ the initial population is created
pob ← ∅
fit ← ∅
for i from 1 to 8t{
     ind ← create_new()
     pob ← pob ∪ {ind}
     fit ← fit ∪ {evaluate(ind)}
}
p_r ← 0.8
while (max(fit)< 4t − 1){
\\ reproduction starts
     if random(0,1) ≤ p_r then{
        j ←random(1,8t)
        ind_j ← the j^th-individual of pob
        list ← guidedreproduction(ind_j)
     else
        i ←random(1,8t)
        j ←random(1,8t) ≠ i
        (ind_i, ind_j) ← the (i^th, j^th)-individuals of pob
        list ← usualreproduction(ind_i, ind_j)
     }
     remove in (pob, fit) those entries corresponding to the less
     size(list) fit individuals
     for i from 1 to size(list){
        pob ← pob ∪ {list(i)}
        fit ← fit ∪ {evaluate(list(i))}
   }
}
List the individuals in pob meeting the optimal fitness, 4t − 1
```

Some auxiliar functions have been used, which we describe now:

– create_new() outputs a binary tuple of length s (s being the dimension of the basis $\mathcal{B}$ of 2-cocycles over G), each bit randomly generated as 0 or 1 with the same probability. A deeper knowledge about the properties of the group G might lead to improved versions of this procedure. As a matter of fact, in

the case of dihedral groups, the number of 1s should be forced to $2t$, as the tables in [5] suggest, since the density of cocyclic Hadamard matrices seems to be maximum with this rate of 1s.
- evaluate(*ind*) measures the fitness of the individual *ind*, that is, counts the number of the Hadamard rows (i.e. those whose summation is zero) in the cocyclic matrix generated by the pointwise product of the matrices related to the 2-cocycles of $\mathcal{B}$ corresponding to the 1s in *ind*. In particular, an individual *ind* gives rise to a cocyclic Hadamard matrix if and only if evaluate(*ind*)= $4t-1$.
- random(min, max) outputs a integer in the range $[min, max]$ randomly generated.
- guidedreproduction(*ind*) applies the heuristic procedure for reproduction on the individual *ind*. The output consists in $4t-1$ new individuals, the $(i+1)^{th}$-row of the i^{th}-individual being Hadamard.
- usualreproduction(ind_i, ind_j) applies the usual crossover operator for reproduction on the individuals ind_i and ind_j. The output consists in 2 new individuals.

4 Guided Reproduction on Dihedral Groups

Denote by D_{4t} the dihedral group $\mathbb{Z}_{2t} \times_\chi \mathbb{Z}_2$ of order $4t$, $t \geq 1$, given by the presentation

$$< a, b | a^{2t} = b^2 = (ab)^2 = 1 >$$

and ordering

$$\{1 = (0,0), a = (1,0), \ldots, a^{2t-1} = (2t-1,0), b = (0,1), \ldots, a^{2t-1}b = (2t-1,1)\}$$

In [8] a representative 2-cocycle f of $[f] \in H^2(D_{4t}, \mathbb{Z}_2) \cong \mathbb{Z}_2^3$ is written interchangeably as a triple (A, B, K), where A and B are the inflation variables and K is the transgression variable. All variables take values ± 1. Explicitly,

$$f(a^i, a^j b^k) = \begin{cases} A^{ij}, & i+j < 2t, \\ A^{ij}K, & i+j \geq 2t, \end{cases} \qquad f(a^i b, a^j b^k) = \begin{cases} A^{ij}B^k, & i \geq j, \\ A^{ij}B^kK, & i < j, \end{cases}$$

Let β_1, β_2 and γ denote the representative 2-cocycles related to $(A, B, K) = (-1, 1, 1), (1, -1, 1), (1, 1, -1)$ respectively.

A basis for 2-coboundaries is described in [5], and consists of the elementary coboundaries $\{\partial_a, \ldots, \partial_{a^{2t-3}b}\}$. This way, a basis for 2-cocycles over D_{4t} is given by $\mathcal{B} = \{\partial_a, \ldots, \partial_{a^{2t-3}b}, \beta_1, \beta_2, \gamma\}$.

We focus in the case $(A, B, K) = (1, -1, -1)$ (that is, $R = \beta_2\gamma$), since computational results in [8,5] suggest that this case contains a large density of cocyclic Hadamard matrices.

Furthermore, as it is pointed out in Theorem 2 of [5], cocyclic matrices over D_{4t} using R are Hadamard matrices if and only if rows from 2 to t are Hadamard. We have updated the genetic algorithm in turn, so that only rows from 2 to t

are used in order to check whether their summations are zero. Accordingly, the fitness of an individual runs through the range $[0, t-1]$.

In order to define the heuristic procedure for reproduction we need to know how the 2-coboundaries in $\mathcal{B}$ have to be combined to form i-paths, $2 \leq i \leq t$. This information is given in Proposition 7 of [5].

Proposition 2. *[5] For $1 \leq i \leq 2t$, a maximal i-walk consists of a maximal subset in*

$$(M_{\partial_1}, \ldots, M_{\partial_{2t}}) \quad or \quad (M_{\partial_{2t+1}}, \ldots, M_{\partial_{4t}})$$

formed from matrices $(\ldots, M_j, M_k, \ldots)$ which are cyclically separated in $i-1$ positions (that is $j \pm (i-1) \equiv k \mod 2t$).

We now have enough information about how to combine 2-coboundaries in $\mathcal{B}$ in order to modify the value of $2c_i - 2I_i$, so that $2c_i - 2I_i = 2t - r_i$, that is, the i^{th}-row of our individual being Hadamard.

Notice that since $r_i = 2(i-1)$ for $2 \leq i \leq t$, the cocyclic Hadamard test reduces to $c_i - I_i = t - i + 1$, for $2 \leq i \leq t$.

We include below a pseudo-code of the `guidedreproduction` procedure described in the section before, particularized to the case of dihedral groups.

```
Input: an individual ind of the population
Output: a list newpob of 4t−1 individuals, the (i+1)^th-row of the
i^th-individual being Hadamard

newpob ← ∅
for i from 2 to t{
    ipaths ← list with the maximal i-paths naturally related to ind
    c ← size of ipaths
    intersec ← intersecting positions of −1s in the i^th-row of ind
    I ← size of intersec
    while c − I ≠ t − i + 1{
       if c − I > t − i + 1{
          ind ← decrease(ipaths, intersec, i − 1, random(1, 2))
       else{
          ind ← increase(ipaths, intersec, i − 1, random(1, 3))
      }
       recompute the values ipaths, c, intersec and I related to ind
   }
    newpob ← newpob ∪ {ind}
}
newpob
```

Some auxiliar functions have been used, which we describe now:

– decrease($ipaths, intersec, i-1, j$) tries to decrease the value $c - I$, that is, size($ipaths$)−size($intersec$). This function acts in a different way, depending on the value of $1 \leq j \leq 2$:

  - decrease($ipaths, intersec, i-1, 1$) outputs an individual ind with exactly size($ipaths$) − 1 i-paths. More concretely, it extends one of the i-paths (say p_1, randomly selected) in $ipath$ to the left, until this i-path is connected to a previously existent i-path, say p_2. There are two possibilities now: if $p_1 \neq p_2$, then p_1 and p_2 have been merged into a solely path. On the contrary, if $p_1 = p_2$, then p_1 has been extended to form a i-cycle. In both cases, we have effectively generated a new individual consisting of size($ipaths$) − 1 i-paths.
  - decrease($ipaths, intersec, i-1, 2$) outputs an individual ind with exactly size($intersec$) + 1 intersections. It suffices to randomly choose a 2-coboundary sharing a negative entry with R in the i^{th}-row, in case that it exists. Otherwise the function

$$\text{decrease}(ipaths, intersec, i-1, 1)$$

    should be called.

– increase($ipaths, intersec, i-1, j$) tries to increase the value $c - I$, that is, size($ipaths$)−size($intersec$). This function acts in a different way, depending on the value of $1 \leq j \leq 3$:

  - increase($ipaths, intersec, i-1, 1$) tries to increase the number of the i-paths in $ipaths$, by splitting an existent i-path into two different i-paths. This is only possible for i-paths consisting of at least three 2-coboundaries. If it is the case, it suffices to delete any 2-coboundary different from the extremes of the i-path. If not, the function

$$\text{increase}(ipaths, intersec, i-1, 1 + random(1, 2))$$

    is called.
  - increase($ipaths, intersec, i-1, 2$) tries to increase the number of the i-paths in $ipaths$, by adding a new i-path in $ipaths$ which does not extend any of the previously existent i-paths. This is only possible if a 2-coboundary exists such that it is not adjacent to any of the i-paths in $ipaths$. If it is not the case, the function

$$\text{increase}(ipaths, intersec, i-1, 2 + (-1)^{random(1,2)})$$

    is called.
  - increase($ipaths, intersec, i-1, 3$) tries to create an individual ind with size($intersec$)−1 intersections. It suffices to randomly delete a 2-coboundary sharing a negative entry with R in the i^{th}-row, in case that it exists. Otherwise the function

$$\text{increase}(ipaths, intersec, i-1, random(1, 2))$$

    is called.

5 Examples and Further Work

All the calculations of this section have been worked out in MATHEMATICA 4.0, running on a *Pentium IV 2.400 Mhz DIMM DDR266 512 MB*.

The table below shows some cocyclic Hadamard matrices over D_{4t} (understood as the pointwise linear combinations of the corresponding 2-cocycles of the basis $\mathcal{B}$ described in the preceding section), and the number of iterations and time required (in seconds) as well. Notice that the number of generations is not directly related to the size of the matrices, because of the randomness inherent in any genetic algorithm.

t	iter.	time	product of generators of 2-cocycles over D_{4t}
2	0	$0''$	$(0,1,0,0,0,0,1,1)$
3	0	$0''$	$(0,1,1,1,0,0,1,0,0,0,1,1)$
4	0	$0''$	$(0,1,0,1,1,1,0,0,1,1,1,0,0,0,1,1)$
5	1	$0.2''$	$(1,1,0,0,1,1,1,1,0,0,1,0,0,0,0,1,0,0,1,1)$
6	1	$0.4''$	$(1,0,1,0,1,1,1,0,0,0,1,0,1,1,0,1,1,0,0,0,0,0,1,1)$
7	2	$2.87''$	$(0,1,1,1,0,0,0,0,0,1,0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,0,1,1)$
8	2	$4.54''$	$(0,1,0,0,1,1,1,0,0,1,0,0,0,0,0,1,1,1,0,1,0,1,1,0,0,0,1,0,1,0,1,1)$
9	11	$51.2''$	$(1,1,0,1,1,1,1,0,1,1,1,1,0,1,0,0,1,0,0,0,1,1,0,1,0,1,1,1,0,1,0,0,0,0,1,1)$
10	8	$65.11''$	$(1,0,0,0,1,0,0,1,1,0,1,1,1,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0,0,0,0,0,0,1,1,1,0,0,1,1)$
11	93	$21'$	$(1,1,0,1,1,0,1,1,0,1,0,1,0,1,0,0,0,0,0,1,1,1,1,1,0,1,1,0,1,1,1,0,0,1,1,1,0,1,1,1,1,0,1,1)$
12	44	$18'44''$	$(0,0,1,1,0,1,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,0,0,1,0,0,1,0,1,0,1,0,1,0,0,1,0,0,1,1)$
13	40	$22'12''$	$(1,1,1,0,1,1,1,1,1,1,0,0,0,1,1,0,1,1,1,1,0,0,1,0,1,0,0,1,0,0,1,1,1,1,1,0,1,1,0,1,0,1,0,0,0,1,0,1,0,0,1,1)$

We have experimented 20 runs for each odd value of $3 \leq t \leq 13$ and for each of the values of the parameter p_r over the range $p_r = i/10$, $0 \leq i \leq 10$, which we can not reproduce here due to the page constraints. All of them found some cocyclic Hadamard matrices. Experimentally, the average time and the average number of required iterations suggest that the optimum value for p_r is 0.8.

Unfortunately, the algorithm has not been able to find cocyclic Hadamard matrices for $t > 13$ due to memory limitations: the computer breaks as soon as 5 hours (or equivalently 3000 generations) are reached.

The authors are convinced that improved versions of the algorithm are still to be implemented (for instance, as soon as a method for simultaneously generating a group of Hadamard rows is described). The work in [1] supports this idea.

References

1. Álvarez, V., Armario, J.A., Frau, M.D., Gudiel, F., Osuna, A.: Rooted trees searching for cocyclic Hadamard matrices over D_{4t}. In: AAECC18 Proceedings. LNCS. Springer, Heidelberg (to appear, 2009)
2. Álvarez, V., Armario, J.A., Frau, M.D., Real, P.: A genetic algorithm for cocyclic Hadamard matrices. In: Fossorier, M.P.C., Imai, H., Lin, S., Poli, A. (eds.) AAECC 2006. LNCS, vol. 3857, pp. 144–153. Springer, Heidelberg (2006)
3. Álvarez, V., Armario, J.A., Frau, M.D., Real, P.: Calculating cocyclic Hadamard matrices in *Mathematica*: exhaustive and heuristic searches. In: Iglesias, A., Takayama, N. (eds.) ICMS 2006. LNCS, vol. 4151, pp. 419–422. Springer, Heidelberg (2006)
4. Álvarez, V., Armario, J.A., Frau, M.D., Real, P.: The homological reduction method for computing cocyclic Hadamard matrices. J. Symb. Comput. (2008) (available online 30-September-2008), doi:10.1016/j.jsc.2007.06.009
5. Álvarez, V., Armario, J.A., Frau, M.D., Real, P.: A system of equations for describing cocyclic Hadamard matrices. Journal of Comb. Des. 16 (4), 276–290 (2008)
6. Baliga, A., Chua, J.: Self-dual codes using image restoration techniques. In: Bozta, S., Sphparlinski, I. (eds.) AAECC 2001. LNCS, vol. 2227, pp. 46–56. Springer, Heidelberg (2001)
7. Flannery, D.L.: Calculation of cocyclic matrices. J. of Pure and Applied Algebra 112, 181–190 (1996)
8. Flannery, D.L.: Cocyclic Hadamard matrices and Hadamard groups are equivalent. J. Algebra 192, 749–779 (1997)
9. Hedayat, A., Wallis, W.D.: Hadamard Matrices and Their Applications. Ann. Stat. 6, 1184–1238 (1978)
10. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
11. Horadam, K.J.: Hadamard matrices and their applications. Princeton University Press, Princeton (2006)
12. Horadam, K.J., de Launey, W.: Cocyclic development of designs. J. Algebraic Combin. 2(3), 267–290 (1993); Erratum: J. Algebraic Combin. (1), 129 (1994)
13. Horadam, K.J., de Launey, W.: Generation of cocyclic Hadamard matrices. In: Computational algebra and number theory. Math. Appl., vol. 325, pp. 279–290. Kluwer Acad. Publ., Dordrecht (1995)
14. Kotsireas, I.S., Pinheiro, G.: A meta-software system for the discovery of Hadamard matrices. In: HPCS 2005 Proceedings, IEEE Computer Society P2343, pp. 17–23 (2005)
15. MacWilliams, F.J., Sloane, N.J.A.: The theory of error-correcting codes. North Holland, New York (1977)