JOURNAL OF COMBINATORIAL THEORY, Series A 27, 289–306 (1979)

An Algorithm for Computing the Automorphism Group of a Hadamard Matrix

JEFFREY S. LEON*

Department of Mathematics, University of Illinois at Chicago Circle, Chicago, Illinois 60608

Communicated by Marshall Hall, Jr.

Received October 3, 1977

An algorithm for computing the automorphism group of a Hadamard matrix is described. It is shown how the algorithm may be modified to determine if two Hadamard matrices are equivalent.

INTRODUCTION

A Hadamard matrix of dimension n is an $n \times n$ matrix H of -1's and 1's such that $HH^T = nI$. The dimension n must be 1, 2, or a multiple of 4. Hadamard matrices of dimension n are known to exist for many, but not all, values of n which are multiples of 4 (see Ref. [1]). Two Hadamard matrices H_1 and H_2 of the same dimension are equivalent if H_2 may be obtained from H_1 by a signed permutation of the rows and columns (precise definitions in Section 2); the signed permutation of rows and columns is called an isomorphism from H_1 to H_2 or, when $H_2 = H_1$, an automorphism of H_1. Under composition, the automorphisms of H_1 form a group called the automorphism group. This group provides a measure of the symmetry of H_1; for example, if it is transitive on rows, any row of H_1 is "equivalent" to any other row.

This article describes an algorithm for computing the automorphism group of a Hadamard matrix. It also shows how the algorithm may be modified to determine if two Hadamard matrices are equivalent. The algorithm has been programmed (in Fortran) and run on a computer. It has proved quite efficient for matrices of moderate dimension (up to 50); probably it can be applied, under appropriate conditions, to matrices of dimension as high as 100. The algorithm yields

* Work partially supported by NSF Grant MCS76-03143. Computing services used in this research were provided by the Computer Center of the University of Illinois at Chicago Circle. Their assistance is gratefully acknowledged.

0097-3165/79/060289–18$02.00/0

(a) the order of the automorphism group,

(b) the orbits of the automorphism group on the rows and columns of the matrix, and

(c) a "canonical" set of signed permutations which generate the automorphism group; modulo sign changes, these permutations form a strong generating set (relative to a suitable base) for the automorphism group acting as a permutation group on the set of rows and columns.

The concepts of base and strong generating set were developed by Sims to facilitate computation in large permutation groups. They are discussed briefly in Section 4; more details may be found in Refs. [3, 4].

This article is intended to be fairly self-contained; however, a familiarity with backtrack search techniques would be helpful in understanding the algorithm. Backtrack methods in general are discussed in Ref. [2].

1. Ordinary and Signed Permutations

This section describes the notation to be used for ordinary (unsigned) and signed permutations.

The symmetric group on a finite set Ω will be denoted by $S(\Omega)$. Permutations will act on the right; thus $(b^{\tau_1})^{\tau_2} = b^{(\tau_1\tau_2)}$. If G is a subgroup of $S(\Omega)$ and $\{b_1, b_2, \ldots, b_k\} \subseteq \Omega$, then $G_{b_1 b_2 \ldots b_k}$ will denote the stabilizer in G of $b_1, b_2, \ldots, b_k$. The orbit of b under G will be written as b^G. The identity permutation will be denoted by ι. A subgroup G of $S(\Omega)$ is semiregular on Ω if $G_b = \langle \iota \rangle$ for every $b \in \Omega$; in this case, $|G|$ divides $|\Omega|$.

Assuming that the set Ω does not itself have a unary operation minus, let $\pm\Omega$ denote the set of symbols $\{\delta b \mid \delta = \pm 1, b \in \Omega\}$. Elements of $\pm\Omega$ may be multiplied on the left by -1 or 1 in the natural way: $\delta_1(\delta_2 b) = (\delta_1\delta_2)b$. We will write b for $(+1)b$ and $-b$ for $(-1)b$. A signed permutation σ on Ω is a mapping from Ω into $\pm\Omega$ such that $b_1{}^\sigma \neq \pm b_2{}^\sigma$ whenever $b_1 \neq b_2$. Such a map σ can be written uniquely as a product $\mu\tau$ with $\tau \in S(\Omega)$ and $\mu(b) = \pm 1$ for all $b \in \Omega$. Composition of signed permutations is defined by $\sigma_1\sigma_2 = (\mu_1\tau_1)(\mu_2\tau_2) = \mu\tau$, where $\tau = \tau_1\tau_2$ and $\mu(b) = \mu_1(b)\,\mu_2(b^{\tau_1})$. The inverse of $\mu\tau$ is $\tilde{\mu}\tau^{-1}$, where $\tilde{\mu}(b) = \mu(b^{\tau^{-1}})$. Under composition, the signed permutations on Ω form a group denoted here by $\Sigma(\Omega)$. This group is a semidirect product of an elementary Abelian group $\Sigma_0(\Omega) = \{\mu\tau \mid \tau = \iota\}$ of order $2^{|\Omega|}$ with a group $S = \{\mu\tau \mid \mu(b) = 1 \text{ for all } b\}$ isomorphic to $S(\Omega)$. The center $Z(\Omega)$ of $\Sigma(\Omega)$ is $\{\pm\iota\}$, where $b^{-\iota} = -b$ for all b. The map $\mu\tau \to \tau$ is a homomorphism from $\Sigma(\Omega)$ onto $S(\Omega)$ with kernel $\Sigma_0(\Omega)$. The image of a subset Q (or element q) of $\Sigma(\Omega)$ under this map will be written $\bar{Q}$ (or $\bar{q}$). Note that $\overline{\mu\tau} = \tau$.

For a fixed positive integer n, let Ω_R be the set of symbols $\{R1, R2, \ldots, Rn\}$

and Ω_C the set of symbols $\{C1, C2, \ldots, Cn\}$. Let $\Omega_{RC} = \Omega_R \cup \Omega_C$. We will think of Ri and Cj as representing the ith row and jth column of an $n \times n$ matrix, respectively. Let $\hat{\Sigma}(\Omega_{RC})$ denote the subgroup of $\Sigma(\Omega_{RC})$ mapping Ω_R onto Ω_R and Ω_C onto Ω_C. If $\sigma \in \hat{\Sigma}(\Omega_{RC})$, then σ_R and σ_C will denote the restrictions of σ to Ω_R and Ω_C, respectively, and $\tilde{\sigma}_R$ and $\tilde{\sigma}_C$ will be the elements of $\hat{\Sigma}(\Omega_{RC})$ defined by $\tilde{\sigma}_R|_{\Omega_R} = \sigma_R$, $\tilde{\sigma}_R|_{\Omega_C} = \iota$, $\tilde{\sigma}_C|_{\Omega_R} = \iota$, $\tilde{\sigma}_C|_{\Omega_C} = \sigma_C$. Note $\sigma = \tilde{\sigma}_R\tilde{\sigma}_C = \tilde{\sigma}_C\tilde{\sigma}_R$.

Now $\hat{\Sigma}(\Omega_{RC})$ operates on the set of $n \times n$ $(-1, 1)$-matrices as follows: If $H = (h_{ij})$, $\sigma = \mu\tau$, $Ri^{\tau^{-1}} = Ri'$, and $Cj^{\tau^{-1}} = Cj''$, then $H^\sigma = (a_{ij})$, where $a_{ij} = \mu(Ri')\,\mu(Cj'')\,h_{i'j''}$. Thus, if $Ri^\sigma = \delta Rl$, $\delta(\text{row } i)$ of H becomes (row l) of H^σ, and likewise for columns. Note that $(H^{\sigma_1})^{\sigma_2} = H^{(\sigma_1\sigma_2)}$ and $H^\sigma = (H^{\tilde{\sigma}_R})^{\tilde{\sigma}_C} = (H^{\tilde{\sigma}_C})^{\tilde{\sigma}_R}$ (i.e., row and column permutations commute).

Finally, we note that $\Sigma(\Omega_C)$ is isomorphic to the group of n-dimensional $(-1, 1)$-monomial matrices. Given $\sigma \in \Sigma(\Omega_C)$, let $\mathscr{M}_C(\sigma) = (a_{ij})$, $a_{ij} = \mu(Ci)$ if $Ci^\tau = Cj$ and $a_{ij} = 0$ otherwise. Then $\mathscr{M}_C(\sigma_1\sigma_2) = \mathscr{M}_C(\sigma_1)\,\mathscr{M}_C(\sigma_2)$, and the map $\sigma \to \mathscr{M}_C(\sigma)$ provides the isomorphism. Similarly, given $\sigma \in \Sigma(\Omega_R)$, let $\mathscr{M}_R(\sigma) = (d_{ij})$, $d_{ij} = \mu(Rj)$ if $Rj^\tau = Ri$ and $d_{ij} = 0$ otherwise. In this case, $\mathscr{M}_R(\sigma_1\sigma_2) = \mathscr{M}_R(\sigma_2)\,\mathscr{M}_R(\sigma_1)$ and the map $\sigma \to \mathscr{M}_R(\sigma)^{-1}$ provides an isomorphism from $\Sigma(\Omega_R)$ onto the group of n-dimensional $(-1, 1)$-monomial matrices. Note that $H^{\tilde{\sigma}_R} = \mathscr{M}_R(\sigma_R)H$ and $H^{\tilde{\sigma}_C} = H\mathscr{M}_C(\sigma_C)$.

2. Automorphisms and Isomorphisms of Hadamard Matrices

An automorphism of an n-dimensional Hadamard matrix H is a signed permutation σ in $\hat{\Sigma}(\Omega_{RC})$ such that $H^\sigma = H$. Under composition, the automorphisms of H form a subgroup of $\hat{\Sigma}(\Omega_{RC})$ called the automorphism group of H and written here as $\mathrm{AUT}(H)$. Alternatively, the automorphism group may be defined to be the set of pairs (P, Q) of $(-1, 1)$-monomial matrices such that $PHQ = H$, composition being defined by $(P, Q)(P', Q') = (P'P, QQ')$. This definition yields an isomorphic group, the isomorphism being given by $\sigma \to (\mathscr{M}_R(\sigma_R), \mathscr{M}_C(\sigma_C))$. The notation of signed permutations will be used here as we shall be interested in the permutation group properties of automorphisms.

Two n-dimensional Hadamard matrices H and H' are equivalent if $H^\sigma = H'$ for some $\sigma \in \hat{\Sigma}(\Omega_{RC})$. Since $H^{\sigma_1} = H^{\sigma_2}$ if and only if σ_1 and σ_2 are in the same right coset of $\mathrm{AUT}(H)$ in $\hat{\Sigma}(\Omega_{RC})$, the number of Hadamard matrices equivalent to H is $2^{2n}(n!)^2/|\,\mathrm{AUT}(H)|$.

An automorphism σ of H is determined uniquely by either of σ_R or σ_C. This is an immediate consequence of the equation $\mathscr{M}_R(\sigma_R)\,H\mathscr{M}_C(\sigma_C) = H$. Let $\mathrm{AUT}_R(H) = \{\sigma_R \mid \sigma \in \mathrm{AUT}(H)\} \subseteq \Sigma(\Omega_R)$ and $\mathrm{AUT}_C(H) = \{\sigma_C \mid \sigma \in \mathrm{AUT}(H)\} \subseteq \Sigma(\Omega_C)$. These groups will be called the row and column automorphism groups of H, respectively. As abstract groups $\mathrm{AUT}_R(H)$ and

$\mathrm{AUT}_C(H)$ are both isomorphic to $\mathrm{AUT}(H)$ (the maps $\sigma \to \sigma_R$ and $\sigma \to \sigma_C$ are isomorphisms); however, they may act very differently as permutation groups; for example, $\mathrm{AUT}_C(H)$ may be transitive on Ω_C while $\mathrm{AUT}_R(H)$ is intransitive on Ω_R.

Let $R_0(H) = \{\sigma \in \mathrm{AUT}(H) \mid \sigma_R \in \Sigma_0(\Omega_R)\}$ and let $C_0(H) = \{\sigma \in \mathrm{AUT}(H) \mid \sigma_C \in \Sigma_0(\Omega_C)\}$. Then $R_0(H)$ and $C_0(H)$ are normal elementary Abelian 2-subgroups of $\mathrm{AUT}(H)$ containing $Z(H) = \{\pm\iota\}$. In matrix notation, $R_0(H)$ corresponds to pairs (P, Q) with P diagonal and $C_0(H)$ to pairs (P, Q) with Q diagonal.

LEMMA 2.1. *Let H be an n-dimensional Hadamard matrix.*

(a) *If $\sigma \in R_0(H)$ and $Cj^\sigma = Cj$ for some j, then $\sigma = \iota$.*

(b) $R_0(H) \cap C_0(H) = Z(H)$.

(c) $\overline{\mathrm{AUT}(H)} \simeq \mathrm{AUT}(H)/Z(H)$.

(d) $\overline{R_0(H)} \simeq R_0(H)/Z(H)$ *restricted to Ω_C is semiregular; hence $|R_0(H)|$ is a power of 2 dividing $2n$.*

Proof. If the hypotheses of (a) hold and $\sigma = \mu\tau$, then $Ri^\tau = Ri$ for all i, $\mu(Cj) = 1$, and $Cj^\tau = Cj$. For $i = 1,\ldots, n$, the i,jth entries of H and H^σ are h_{ij} and $\mu(Ri)\, h_{ij}$; equating these gives $\mu(Ri) = 1$ for all i. Hence $\sigma_R = \iota$, and part (a) holds.

If $\sigma \in R_0(H) \cap C_0(H)$, either $C1^\sigma = C1$ or $C1^{-\sigma} = C1$. Thus, by (a), either $\sigma = \iota$ or $\sigma = -\iota$, and part (b) holds. Part (c) follows from (b) since the kernel in $\mathrm{AUT}(H)$ of the map $\sigma \to \bar{\sigma}$ is $R_0(H) \cap C_0(H)$. Part (d) is an immediate consequence of (a).

3. Structures Invariant under Automorphisms and Isomorphisms

An automorphism σ of an n-dimensional Hadamard matrix H is determined uniquely by σ_R in $\Sigma(\Omega_R)$; however, as $|\Sigma(\Omega_R)| = 2^n n!$, it is not feasible, at least when $n > 8$, to examine each element of $\Sigma(\Omega_R)$ in searching for automorphisms of H. Given relatively small subsets $B = \{Ri_1, Ri_2, \ldots, Ri_k\}$ of Ω_R, we need criteria for showing that some maps from B into $\pm\Omega_R$ cannot be extended to automorphisms or, alternatively, that certain maps from B into Ω_R cannot be extended to permutations τ in $\hat{S}(\Omega_{RC})$ such that $\mu\tau$ is an automorphism for some μ.

If only rows $i_1, i_2, \ldots, i_k$ and their images under the map are examined, the smallest value of k for which maps may be eliminated is 4. This follows from the fact that any three rows may be transformed to any other three rows by a signed permutation of rows and columns; in fact, both sets of three rows may be transformed to

$$\left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & - & - & \cdots & - & - & - & \cdots & - \\ 1 & 1 & \cdots & 1 & - & - & \cdots & - & 1 & 1 & \cdots & 1 & - & - & \cdots & - \end{array}\right].$$

$n/4 \qquad n/4 \qquad n/4 \qquad n/4$

We will see that any four rows may be transformed to the form M_l below for one value of l, and only one, with $n/8 \leqslant l \leqslant n/4$.

$$M_l = \left[\begin{array}{ccccc|ccccc|ccccc|ccccc} 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & - & \cdots & - & - & \cdots & - & - & \cdots & - \\ 1 & \cdots & 1 & 1 & \cdots & 1 & - & \cdots & - & - & \cdots & - & 1 & \cdots & 1 & 1 & \cdots & 1 & - & \cdots & - \\ 1 & \cdots & 1 & - & \cdots & - & 1 & \cdots & 1 & - & \cdots & - & 1 & \cdots & 1 & - & \cdots & - & 1 & \cdots & - \end{array}\right]$$

$$l \quad \frac{n}{4}-l \quad \frac{n}{4}-l \quad l \quad \frac{n}{4}-l \quad l \quad l \quad \frac{n}{4}-l$$

Orthogonality among any four rows of H implies that, for some q with $0 \leqslant q \leqslant n/4$, the $4 \times n$ submatrix formed by these rows has q columns of each of the four forms

$$\pm\begin{pmatrix}1\\1\\1\\1\end{pmatrix}, \quad \pm\begin{pmatrix}1\\1\\-\\-\end{pmatrix}, \quad \pm\begin{pmatrix}1\\-\\1\\-\end{pmatrix}, \quad \text{and} \quad \pm\begin{pmatrix}1\\-\\-\\1\end{pmatrix}$$

and $n/4 - q$ columns of each of the four forms

$$\pm\begin{pmatrix}1\\1\\1\\-\end{pmatrix}, \quad \pm\begin{pmatrix}1\\1\\-\\1\end{pmatrix}, \quad \pm\begin{pmatrix}1\\-\\1\\1\end{pmatrix}, \quad \text{and} \quad \pm\begin{pmatrix}1\\-\\-\\-\end{pmatrix}.$$

This integer q is invariant under signed permutation of columns and ordinary permutation of rows; negating a single row changes q to $n/4 - q$. Thus four rows may be transformed to M_l if and only if $l = q$ or $n/4 - q$ (M_q may be transformed to $M_{n/4-q}$ by negating the last row and permuting the columns); it follows that four rows may be transformed to M_l for a unique l with $n/8 \leqslant l \leqslant n/4$.

DEFINITION. If H is an n-dimensional Hadmaard matrix and if u, v, w, and x are distinct elements of $\{1,\ldots, n\}$, then $\Phi_H(Ru, Rv, Rw, Rx) = |\{j \mid h_{uj} = h_{vj} = h_{wj} = h_{xj}\}|$.

Thus Φ gives the integer q associated with rows u, v, w, and x. The argument of the last paragraph shows:

PROPOSITION 3.1. *If* $\sigma = \mu\tau \in \mathrm{AUT}(H)$, *then* $\Phi_H(Ru^\tau, Rv^\tau, Rw^\tau, Rx^\tau) = \Phi_H(Ru, Rv, Rw, Rx)$ *or* $n/4 - \Phi_H(Ru, Rv, Rw, Rx)$.

Once Φ_H has been computed for all four element sets, it is easy to obtain criteria for eliminating maps from $B = \{Ri_1, \ldots, Ri_k\}$, $k \leqslant 3$, into Ω_R. Let $p = n/8$ or $(n + 4)/8$, whichever is an integer.

DEFINITION. For $k = 1, 2, 3$, $\Psi_H^{(k)}(Ri_1, \ldots, Ri_k) = (\gamma_0, \ldots, \gamma_{n/4})$, where $\gamma_l = |\{\{u, v, w, x\} \mid \{i_1, \ldots, i_k\} \subseteq \{u, v, w, x\}$ and $\Phi_H(Ru, Rv, Rw, Rx) = l$ or $n/4 - l\}|$.

From Proposition 3.1, it follows immediately that:

PROPOSITION 3.2. *If* $\sigma = \mu\tau \in \mathrm{AUT}(H)$, *then* $\Psi_H^{(k)}(Ri_1^\tau, \ldots, Ri_k^\tau) = \Psi_H^{(k)}(Ri_1, \ldots, Ri_k)$.[1]

Naturally, Propositions 3.1 and 3.2 have analogs in the case that σ is an isomorphism between two Hadamard matrices H_1 and H_2.

4. BASES AND STRONG GENERATING SETS FOR PERMUTATION GROUPS

To facilitate computation in groups of ordinary (unsigned) permutations, Sims introduced the concepts of base and strong generating set. These concepts will be summarized here and then applied to the group $\overline{\mathrm{AUT}(H)}$ acting on Ω_{RC}.

Let G be a permutation group on a finite set Ω. A subset $\{d_1, d_2, \ldots, d_m\}$ of Ω is a base for G if $G_{d_1 d_2 \ldots d_m} = \langle \iota \rangle$. A generating set S for G is a strong generating set relative to the ordered base $(d_1, d_2, \ldots, d_m)$ if $G_{d_1 \ldots d_{i-1}} = \langle S \cap G_{d_1 \ldots d_{i-1}} \rangle$ for $i = 1, 2, \ldots, m$, that is, if S contains generating sets for each of the groups $G, G_{d_1}, \ldots, G_{d_1 \ldots d_{m-1}}$. The $(i-1)$-point stabilizer $G_{d_1 \ldots d_{i-1}}$ will be denoted here by $G^{(i)}$.

Given a base and strong generating set for G, it is easy to compute $\Delta_i = d_i^{G^{(i)}}$ for $i = 1, 2, \ldots, m$ (Section 5 contains an orbit algorithm). The set Δ_i is called the ith basic orbit of G. It is also possible to construct a set V_i of right coset representatives for $G^{(i+1)}$ in $G^{(i)}$; in fact, V_i is in one-to-one correspondence with Δ_i, the correspondence being obtained by choosing, for each $c \in \Delta_i$, a permutation ν in $G^{(i)}$ with $d_i^\nu = c$. Every element of G may be written uniquely in the form $\nu_m \nu_{m-1} \cdots \nu_1$ with $\nu_i \in V_i$. It follows that $|G| = \prod_{i=1}^m |\Delta_i|$.

Now suppose that the points of $\Omega - \{d_1, d_2, \ldots, d_m\}$ are labeled $d_{m+1}, \ldots, d_n$.

[1] Let Z denote the set of integers, $m = n/4 - p + 1$, and $Z^m = Z \times Z \times \cdots \times Z$ (m factors). It is convenient to replace Ψ_H by $\theta\Psi_H$, where θ is some (presumably random) function from Z^m into Z. Note Proposition 3.2 remains valid.

Define the ordering $\ll$ on Ω by $d_1 \ll d_2 \ll \cdots \ll d_n$. The elements g of G may be ordered lexicographically according to $(d_1^g, d_2^g, \ldots, d_m^g)$. Thus $g \ll h$ if, for some i, $d_i^g \ll d_i^h$ but $d_j^g = d_j^h$ whenever $j < i$. With this ordering, any element of $G^{(j)}$ precedes any element of $G - G^{(j)}$.

Relative to the ordering of Ω, G has a unique generating set $g_1, g_2, \ldots, g_q$ with the property that g_i is the first element of $G - \langle g_1, g_2, \ldots, g_{i-1} \rangle$ for $i = 1, 2, \ldots, q$ (by convention, the empty set generates $\langle \iota \rangle$). This unique generating set will be referred to here as the canonical generating set for G. The canonical generating set is a strong generating set, for if $\langle g_1, \ldots, g_i \rangle \subsetneq G^{(j)}$, then g_{i+1} must lie in $G^{(j)}$ as it is the first element of $G - \langle g_1, \ldots, g_i \rangle$.

In some applications, the first N canonical generators $g_1, g_2, \ldots, g_N$ will be known while g_{N+1} is being searched for; in this situation, the images of the base under g_{N+1} are restricted by:

PROPOSITION 4.1. *Let $g_1, g_2, \ldots, g_q$ be the canonical generating set for G on Ω (relative to the base $d_1, \ldots, d_m$ and the ordering $\ll$), and let $1 \leqslant N \leqslant q - 1$. Set $K = \langle g_1, \ldots, g_N \rangle$ and $K^{(j)} = K \cap G^{(j)}$. If d_r is the first base point moved by g_{N+1}, then:*

(a) *$d_r^{g_{N+1}}$ is the first point (relative to $\ll$) in its K-orbit.*

(b) *For $l = r + 1, \ldots, m$ and $j = r, \ldots, l - 1$, $d_l^{g_{N+1}} \gg d_j^{g_{N+1}}$ whenever $d_l \in d_j^{K^{(j)}}$.*

(c) *If $j > r$ and $\Gamma \supseteq \bigcup_{i=r}^{j-1} d_i^K$, then $d_j^{g_{N+1}}$ is first in its K_Γ orbit ($K_\Gamma =$ pointwise stabilizer of Γ).*

Proof. The ordering chosen for G implies that $K = K^{(r)} \subseteq G^{(r)}$ and $K^{(j)} = G^{(j)}$ for $j > r$. Since g_{N+1} is first in $G - K$, it is first in its left K-coset $g_{N+1}K$ and first in its right K-coset Kg_{N+1}.

If (a) fails, then $d_r^{g_{N+1}k} \ll d_r^{g_{N+1}}$ for some $j \in K$. As $d_i^{g_{N+1}k} = d_i^{g_{N+1}} = d_i$ for all $i < r$, $g_{N+1}k$ precedes g_{N+1}, contrary to g_{N+1} being first in its left K-coset.

If (b) fails, $d_l^{g_{N+1}} \ll d_j^{g_{N+1}}$ and $d_l = d_j^k$ for some $k \in K^{(j)}$. Then $d_j^{kg_{N+1}} \ll d_j^{g_{N+1}}$. As $d_i^{kg_{N+1}} = d_i^{g_{N+1}}$ for all $i < j$, kg_{N+1} precedes g_{N+1}, contrary to g_{N+1} being first in its right K-coset.

If (c) fails, $d_j^{g_{N+1}k} \ll d_j^{g_{N+1}}$ for some $k \in K_\Gamma$. Since $d_i^{g_{N+1}} \in \Gamma$ for $i = r, \ldots, j - 1$, $d_i^{g_{N+1}k} = d_i^{g_{N+1}}$ and $g_{N+1}k$ precedes g_{N+1}, contrary to g_{N+1} being first in its left K-coset.

Note that tests (a), (b), and (c) above require knowledge of g_{N+1} only on the base or an initial segment of the base.

The concepts of base and strong generating set are most useful when the size of the base is small compared to the degree. Although such bases exist for many interesting permutation groups, there is in general no guarantee that a small base will exist. However, when H is an n-dimensional Hadamard matrix, we will see that $\overline{\mathrm{AUT}(H)}$ on Ω_{RC} always has a base with at most

$2 \log_2 n + 2$ elements; moreover, we will obtain an easy method for constructing such a base.

DEFINITION. Let H be an n-dimensional Hadamard matrix. A *row base* for H is a subset $\{b_1, \ldots, b_k\}$ of $\{1, 2, \ldots, n\}$ such that $(h_{b_1 i}, h_{b_2 i}, \ldots, h_{b_k i}) \neq \pm(h_{b_1 j}, h_{b_2 j}, \ldots, h_{b_k j})$ whenever $i \neq j$.

Any row base must have at least $\log_2 n + 1$ elements (there must exist $2n$ distinct k-vectors with entries ± 1). On the other hand, any $(n/2 + 1)$-element subset is a row base. For $n \geq 8$, Proposition 4.2 will show that smaller row bases can always be found. Note that, if $\{b_1, \ldots, b_k\}$ is a row base and $\sigma \in \hat{\Sigma}(\Omega_{RC})$ with $Rb_j^{\sigma} = \pm Ra_j$, then $\{a_1, \ldots, a_k\}$ is a row base of H^{σ}.

PROPOSITION 4.2. *Let H be an n-dimensional Hadamard matrix. Set $\gamma_1 = n(n-1)/2$ and*

$$\gamma_{i+1} = \left[\frac{(n/2 - i)\gamma_i}{n - i}\right]$$

for $i = 1, 2, \ldots$ (brackets denote greatest integer). Then H has a k-element row base for any $k \leq n/2$ with $\gamma_k = 0$.

In particular, H has a row base with at most $2 \log_2 n + 1$ elements.

Proof. Choose b_1 arbitrarily and normalize H so that $h_{b_1 j} = 1$ for all j. Then $\{b_1, \ldots, b_k\}$ will be a row base if $(h_{b_2 j_1}, \ldots, h_{b_k j_1}) \neq (h_{b_2 j_2}, \ldots, h_{b_k j_2})$ whenever $j_1 < j_2$.

Choose $b_2, b_3, \ldots$ inductively as follows: Once $b_2, \ldots, b_i$ have been chosen, set $S_i = \{(j_1, j_2) \mid 1 \leq j_1 < j_2 \leq n, \ h_{b_l j_1} = h_{b_l j_2} \text{ for } l = 2, \ldots, i\}$. For $b \in \{1, 2, \ldots, n\} - \{b_1, \ldots, b_i\}$, set $c_b = |\{(j_1, j_2) \in S_i \mid h_{bj_1} = h_{bj_2}\}|$ and choose b_{i+1} so that $c_{b_{i+1}} = \min\{c_b\}$.

Now $\{b_1, \ldots, b_k\}$ will be a row base provided $S_k = \phi$; thus it will suffice to prove by induction that $|S_i| \leq \gamma_i$. Clearly this is true (with equaltiy) when $i = 1$; assume true for i. Any pair (j_1, j_2) of columns in S_i agree in $n/2$ rows including the i rows $b_1, \ldots, b_i$. Thus there are $(n/2 - i)|S_i|$ triples (j_1, j_2, b) with $(j_1, j_2) \in S_i$, $b \neq b_1, \ldots, b_i$, and $h_{bj_1} = h_{bj_2}$. As there are $n - i$ choices for b, for some b there are at most

$$\left[\frac{(n/2 - i)|S_i|}{n - i}\right]$$

triples. Thus

$$|S_{i+1}| \leq \left[\frac{(n/2 - i)|S_i|}{n - i}\right] \leq \left[\frac{(n/2 - i)\gamma_i}{n - i}\right] = \gamma_{i+1}.$$

As $\gamma_1 < n^2/2$ and $\gamma_{i+1} < \gamma_i/2$, γ_k must be zero whenever $k \geq 2 \log_2 n$.

For $8 \leqslant n \leqslant 100$, the minimal row-base size k guaranteed by Proposition 4.2 is as follows:

n	k
8	4
12–16	6
20	7
24–28	8
32–40	9
44–60	10
64–76	11
80–100	12

PROPOSITION 4.3. *Let $\{b_1,\ldots, b_k\}$ be a row base for the Hadamard matrix H. If $\sigma \in \mathrm{AUT}(H)$ and $Rb_i^\sigma = Rb_i$ for $i = 1,\ldots, k$, then $\sigma = \iota$.*

Proof. If $Cj^\sigma = \delta Cl$, $\delta = \pm 1$, then $\delta(h_{b_1 j},\ldots, h_{b_k j}) = (h_{b_1 l},\ldots, h_{b_k l})$, impossible by the definition of a row base unless $l = j$ and $\delta = 1$. Thus $\sigma_C = \iota$. It follows that $\sigma = \iota$.

A row base for H need not form a base for $\overline{\mathrm{AUT}(H)}$ on Ω_{RC} (the stabilizer of the row base includes $\overline{R_0(H)}$); however, a row base plus any column does form a base for $\overline{\mathrm{AUT}(H)}$.

PROPOSITION 4.4. *If $\{b_1,\ldots, b_k\}$ is a row base for the n-dimensional Hadamard matrix H and $j \in \{1, 2,\ldots, n\}$, then $\{Rb_1, Rb_2,\ldots, Rb_k, Cj\}$ is a base for $\overline{\mathrm{AUT}(H)}$ on Ω_{RC}.*

Proof. Let $\sigma = \mu\tau \in \mathrm{AUT}(H)$ with $Rb_i^\tau = Rb_i$ for $i = 1,\ldots, k$ and $Cj^\tau = Cj$ (recall $\tau = \bar{\sigma}$). Equating the $b_i j$th entries of H and H^σ yields $h_{b_i j} = \mu(Rb_i)\,\mu(Cj)\,h_{b_i j}$. Thus $\mu(Rb_i) = \mu(Cj)$ for $i = 1,\ldots, k$. Let $\delta = \mu(Cj)$. Then $Rb_i^\sigma = \delta Rb_i$ for $i = 1,\ldots, k$, and $Rb_i^{\delta\sigma} = Rb_i$. By Proposition 4.3, $\delta\sigma = \iota$. Thus $\sigma = \delta\iota$ and $\bar{\sigma} = \iota$.

COROLLARY 4.5. *Let M be a subgroup of $\mathrm{AUT}(H)$ with $\overline{M}_{Rb_1,\ldots, Rb_k} = \overline{\mathrm{AUT}(H)}_{Rb_1,\ldots, Rb_k}$, $\{b_1,\ldots, b_k\}$ a row base. Then*

(a) *M is an elementary Abelian 2-group,*

(b) *M is semiregular on Ω_C.*

Proof. By Proposition 4.3, M acts faithfully on $\{Rb_1,\ldots, Rb_k\}$; however, M merely induces sign changes there; hence (a) holds. Part (b) follows immediately from Proposition 4.4.

5. Some Preliminary Algorithms

This section contains several short algorithms which will be referred to in the automorphism group algorithm. In Fortran, these short algorithms are coded as subroutines. All variables except the parameters (the quantities in parentheses following the algorithm name) are local to the particular algorithm.

The first algorithm will be used for computing the orbits of a permutation group on a set. The orbits induce a partition of the set. In general, a partition of a set Ω, linearly ordered by $<$, may be represented by a function $P: \Omega \to \Omega$ such that for all $\omega \in \Omega$

(i) $P(\omega)$ is in the block of ω,

(ii) $P(\omega) \leqslant \omega$, and $P(\omega) < \omega$ if ω is not first in its block.

Set $P^1(\omega) = P(\omega)$, $P^{k+1}(\omega) = P(P^k(\omega))$, and $P^\infty(\omega) = \lim_{k\to\infty} P^k(\omega)$. For any $\omega \in \Omega$, there exists $i < |\Omega|$ with $P^{i+1}(\omega) = P^i(\omega)$; $P^\infty(\omega)$ may be computed by finding the first such i and setting $P^\infty(\omega) = P^i(\omega)$. Note $P^\infty(\omega)$ is the first element of the block of ω, and ω_1 and ω_2 are in the same block if and only if $P^\infty(\omega_1) = P^\infty(\omega_2)$. Often it is convenient to assume, in addition to (i) and (ii),

(iii) $P^2 = P$ (P is idempotent).

If P is idempotent, $P^\infty = P$; P may be made idempotent by replacing P by P^∞.

Algorithm 1—Join(Ω, P, τ). Initially P represents a partition of $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ and $\tau \in S(\Omega)$; on termination, P is idempotent and represents the join of the initial partition and the partition induced by the cycles of τ.

1. For $j = 1, 2, \ldots, n$:
   (a) Set $\rho_1 = P^\infty(\omega_j)$.
   (b) Set $\rho_2 = P^\infty(\omega_j{}^\tau)$.
   (c) If $\rho_1 < \rho_2$, set $P(\rho_2) = \rho_1$;
   If $\rho_2 < \rho_1$, set $P(\rho_1) = \rho_2$.
2. For $j = 1, 2, \ldots, n$, set $P(\omega_j) = P(P(\omega_j))$.

Performing step 1 for a fixed j puts ω_j and $\omega_j{}^\tau$ into the same block of the partition represented by P. When step 1 is finished, P represents the join of the two partitions; step 2 makes P idempotent. Note that, even if the initial P is idempotent, P will usually become nonidempotent during the course of step 1.

By repeated application of Algorithm 1, the partition induced by the orbits of $\langle \tau_1, \tau_2, \ldots, \tau_m \rangle$ may be found.

The second algorithm constructs a row base for a Hadamard matrix using the method given in the proof of Proposition 4.2.

ALGORITHM 2—ROWBASE(n, H, k, B). Initially, $H = (h_{ij})$ is an n-dimensional Hadamard matrix, $k \geqslant 0$, and $B = \{b_1, \ldots, b_k\}$ is a distinct subset of $\{1, \ldots, n\}$. The algorithm increases k and extends B so that, upon termination, B is a row base for H. If $k \leqslant 3$ initially, upon termination k will satisfy the bounds of Proposition 4.2.

1. If $k < 3$:
   (a) Choose $b_{k+1}, \ldots, b_3$ to be distinct elements of $\{1, 2, \ldots, n\} - \{b_1, \ldots, b_k\}$.
   (b) Set $k = 3$.
2. For $j = 1, 2, \ldots, n$:
   (a) Set $\delta_j = h_{b_1 j}$.
   (b) For $i = 1, 2, \ldots, n$, set $h_{ij} = \delta_j h_{ij}$.
3. Set $d = 0$.
   For $i = 1, 2, \ldots, n$, set $c_i = 0$.
4. For each pair (j_1, j_2) with $1 \leqslant j_1 < j_2 \leqslant n$ and $(h_{b_2 j_1}, \ldots, h_{b_k j_1}) = (h_{b_2 j_2}, \ldots, h_{b_k j_2})$:
   (a) Set $d = d + 1$.
   (b) For each $i = 1, 2, \ldots, n$ with $h_{ij_1} = h_{ij_2}$, set $c_i = c_i + 1$.
5. If $d \neq 0$:
   (a) Set $k = k + 1$.
   (b) Choose b_k so that $c_{b_k} = \min\{c_i \mid 1 \leqslant i \leqslant n\}$.
   (c) If $c_{b_k} \neq 0$, go to step 3.
6. For $i, j = 1, 2, \ldots, n$, set $h_{ij} = \delta_j h_{ij}$.

Step 1 is not strictly necessary unless $k = 0$; however, it saves time as in step 5 the c_i's would be equal if $k < 3$. In steps 4 and 5, i may be restricted to $\{1, \ldots, n\} - \{b_1, \ldots, b_k\}$.

The next two algorithms treat the problem of transforming one $(-1, 1)$-matrix to another by signed permutation of rows only or columns only. If $X = (x_{ij})$ is $n \times m$ and $\sigma = \mu\tau \in \Sigma(\Omega_R)$, let $X^\sigma = (z_{ij})$, $z_{ij} = \mu(Ri')\, x_{i'j}$, where $Ri'^{\tau^{-1}} = Ri'$. This is analogous to the definition in Section 1 except that σ is in $\Sigma(\Omega_R)$ rather than $\hat{\Sigma}(\Omega_{RC})$; thus only rows are permuted. In a similar manner, we obtain for $\sigma \in \Sigma(\Omega_C)$ an action on columns only.

ALGORITHM 3—ROWPERM($n, m, X, Y, \sigma, \lambda$). $X = (x_{ij})$ and $Y = (y_{ij})$ are $n \times m$ $(-1, 1)$-matrices. The algorithm produces $\sigma \in \Sigma(\Omega_R)$ with $X^\sigma = Y$ if σ exists; λ is set to 1 if σ exists and to 0 otherwise.

1. Set $i = 1$.
2. For $l = 1, 2, \ldots, n$, set $\alpha_l = 0$.
3. For each $l = 1, 2, \ldots, n$ with $\alpha_l = 0$:
   (a) Set $\delta = x_{i1} y_{l1}$.
   (b) If $\delta(x_{i2}, \ldots, x_{im}) = (y_{l2}, \ldots, y_{lm})$, set $t = l$ and go immediately to step 5 (any remaining values of l are disregarded).
4. Set $\lambda = 0$; the algorithm terminates.
5. Set $Ri^\sigma = \delta Rt$.
   Set $\alpha_t = 1$.
6. If $i < n$, set $i = i + 1$ and go back to step 3.
   If $i = n$, set $\lambda = 1$; the algorithm terminates.

Naturally the above algorithm can be modified to find column rather than row permutations. However, when X and Y are $k \times n$ matrices with k small (at most 10 to 15) and when the algorithm is applied repeatedly with the same X, the procedure below is much faster. Steps 2, 3, and 4 below need not be repeated when the algorithm is applied a second or subsequent time with the same X.

ALGORITHM 4—COLPERM($k, n, X, Y, \sigma, \lambda$). $X = (x_{ij})$ and $Y = (y_{ij})$ are $k \times n$ $(-1, 1)$-matrices such that any two column vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ of X satisfy $\mathbf{x}_1 \neq \pm \mathbf{x}_2$. The algorithm produces $\sigma \in \Sigma(\Omega_C)$ with $X^\sigma = Y$ if σ exists; λ is set to 1 if σ exists and to 0 otherwise.

1. Set $\lambda = 0$, $A = 2^k - 1$.
2. For $j = 1, 2, \ldots, n$, set $T(j) = \sum_{i=1}^{k} \delta_{x_{ij}1} 2^{i-1}$.
   (δ_{uv} denotes 1 if $u = v$ and 0 otherwise).
3. For $l = 0, 1, \ldots, 2^k - 1$, set $V(l) = 0$.
4. For $j = 1, 2, \ldots, n$, set $V(T(j)) = j$ and $V(A - T(j)) = -j$.
5. For $j = 1, 2, \ldots, n$:
   (a) Set $U(j) = \sum_{i=1}^{k} \delta_{y_{ij}1} 2^{i-1}$.
   (b) If $V(U(j)) = 0$:
      (i) Set $j_0 = j - 1$.
      (ii) Go immediately to step 8 (any remaining values of j are disregarded).
   (c) Set $a_j = |V(U(j))|$ and $\eta_j = V(U(j))/a_j$.
   (d) Set $V(U(j)) = 0$ and $V(A - U(j)) = 0$.

6. Set $\lambda = 1$.
   For $j = 1, 2, \ldots, n$, set $Ca_j^{\sigma} = \eta_j C_j$.
7. Set $j_0 = n$.
8. For $j = 1, 2, \ldots, j_0$, set $V(U(j)) = \eta_j a_j$ and $V(A - U(j)) = -\eta_j a_j$.

If H is a Hadamard matrix of dimension n and $\{i_1, \ldots, i_k\} \subseteq \{\pm 1, \pm 2, \ldots, \pm n\}$, let $(H)^{i_1 \cdots i_k}$ denote the $k \times n$ matrix whose lth row is $i_l/|\, i_l \,|$ times row $|\, i_l \,|$ of H. In the next section, Algorithm 4 will be applied to $k \times n$ matrices of this type with k the row-base size. In view of Proposition 4.2 and Algorithm 2, we may assume that k is sufficiently small to permit use of Algorithm 4.

6. The Automorphism Group Algorithm

This section contains the algorithm for computing the automorphism group of an n-dimensional Hadamard matrix H. There are a number of variations on the basic algorithm; several are discussed later in this section.

Starting with a (possibly empty) subset $B = \{b_1, \ldots, b_k\}$ of $\{1, \ldots, n\}$, we apply Algorithm 2 to extend B to a row base $\{b_1, \ldots, b_k\}$ for H.[1] By Proposition 4.4, $\{Rb_1, \ldots, Rb_k, C1\}$ is a base for $\overline{\mathrm{AUT}(H)}$ on Ω_{RC}. After numbering the points of $\{1, \ldots, n\} - \{b_1, \ldots, b_k\}$ as $b_{k+1}, \ldots, b_n$, we order Ω_{RC} by $Rb_1 \lessdot \cdots \lessdot Rb_k \lessdot C1 \lessdot Rb_{k+1} \lessdot \cdots \lessdot Rb_n \lessdot C2 \lessdot \cdots \lessdot Cn$. Set $\mathrm{NEXT}(0) = b_1$, $\mathrm{NEXT}(b_i) = b_{i+1}$ for $i = 1, \ldots, n-1$, $\mathrm{NEXT}(b_n) = 0$. As $\overline{\mathrm{AUT}(H)} \simeq \mathrm{AUT}(H)/Z(H)$, $\mathrm{AUT}(H)$ has a unique generating set $\{\sigma_1, \ldots, \sigma_N\}$ (called the canonical generating set) such that

(i) $\sigma_1 = -\iota$,

(ii) $C1^{\sigma_i} = Cx_i$ (rather than $-Cx_i$) for some x_i, $2 \leqslant i \leqslant N$,

(iii) $\{\bar{\sigma}_2, \ldots, \bar{\sigma}_N\}$ is the canonical generating set for $\overline{\mathrm{AUT}(H)}$ on Ω_{RC} (relative to $\lessdot$).

The algorithm uses the technique of backtrack searching; for a general description of backtrack search algorithms, see Ref. [2]. Following the statement of the algorithm is a discussion of the steps. The algorithm produces

(1) the canonical generating set $\{\sigma_1, \ldots, \sigma_N\}$ for $\mathrm{AUT}(H)$,

(2) an idempotent function P: $\Omega_{RC} \to \Omega_{RC}$ which describes the partition of Ω_{RC} induced by the orbits of $\overline{\mathrm{AUT}(H)}$,

(3) the basic orbit lengths $L_1, \ldots, L_{k+1}$ for $\overline{\mathrm{AUT}(H)}$ on Ω_{RC}. Note $|\, \mathrm{AUT}(H) \,| = 2 \prod_{j=1}^{k+1} L_j$.

[1] For large matrices, the algorithm performs with much greater efficiency if the initial segment B of the row base is chosen so that the functions Φ_H and $\Psi_H^{(k)}$ attain unusual values on B.

MAIN ALGORITHM

A. *Initializations*

1. Set $N = 1$;
   For $j = 1, 2, \ldots, n$, set $Rj^{\sigma_1} = -Rj$ and $Cj^{\sigma_1} = -Cj$.
2. For $j = 1, 2, \ldots, n$, set $P(Rj) = Rj$, $P(Cj) = Cj$, and $P_0(Cj) = Cj$.
3. For $i = 1, 2, \ldots, k + 1$, set $L_i = 1$.
4. For all u, v with $1 \leqslant u < v \leqslant k$, set $e_{uv} = 0$.
5. Set $r = k + 1$.

B. *Backtrack Search (Images of $Rb_1, \ldots, Rb_k$ under Next Canonical Generator)*

1. Set $i = 1$.
2. Set $a_i = 0$.
3. Set $a_i = \text{NEXT}(a_i)$.
4. If $a_i = 0$, then:
   (a) If $i = 1$, algorithm terminates.
   (b) If $i > 1$, set $i = i - 1$ and go back to step 3.
5. If $a_i = a_j$ for some j with $1 \leqslant j < i$, go back to step 3.
6. If $i < r$ and $a_i \neq b_i$, set $r = i$.
7. Go back to step 3 if any of the four conditions below hold:
   (α) $i = r$ and $P(Ra_i) \neq Ra_i$,
   (β) for some j with $r \leqslant j < i$ and $e_{ji} = 1$, $Ra_i \ll Ra_j$,
   (γ) for some u, v, w with $1 \leqslant u < v < w < i$, $\Phi_H(Ra_u, Ra_v, Ra_w, Ra_i) \neq \Phi_H(Rb_u, Rb_v, Rb_w, Rb_i)$ and $\Phi_H(Ra_u, Ra_v, Ra_w, Ra_i) \neq n/4 - \Phi_H(Rb_u, Rb_v, Rb_w, Rb_i)$,
   (δ) $\Psi_H^{(1)}(Ra_i) \neq \Psi_H^{(1)}(Rb_i)$ or $\Psi_H^{(2)}(Ra_u, Ra_i) \neq \Psi_H^{(2)}(Rb_u, Rb_i)$ for some u with $1 \leqslant u < i$ or $\Psi_H^{(3)}(Ra_u, Ra_v, Ra_i) \neq \Psi_H^{(3)}(Rb_u, Rb_v, Rb_i)$ for some u, v with $1 \leqslant u < v < i$.
8. If $i < k$, set $i = i + 1$ and go back to step 2.

C. *Backtrack Search Continued (Image of $C1$ under Next Canonical Generator)*

1. Set $j = 1$ if $r \leqslant k$; set $j = 2$ if $r = k + 1$.
2. If $P_0(Cj) \neq Cj$, go to step 8.
3. Set $\eta_l = h_{b_{i1}} h_{a_{ij}}$ for $l = 1, 2, \ldots, k$.

4. Apply Algorithm 4—COLPERM(k, n, $(H)^{b_1,\ldots,b_k}$,$(H)^{\eta_1 a_1,\ldots,\eta_k a_k}$, σ_C , λ).[2]
   If $\lambda = 0$, go to step 8.
5. Apply Algorithm 3—ROWPERM(n, n, H^{ϑ_C}, H, σ_R , λ).
   If $\lambda = 0$, go to step 8.
6. Go to part D (part D adds σ to the generating set and returns control to step 7).
7. If $r \leqslant k$, set $i = r$ and go back to step B.3.
8. If $j < n$, set $j = j + 1$ and go back to step 2.
   If $j = n$, go back to step B.3.

D. *Addition of the Next Canonical Generator*

1. Set $N = N + 1$.
   For $l = 1, 2,\ldots, n$, set $Rl^{\sigma N} = Rl^{\sigma}$ and $Cl^{\sigma N} = Cl^{\sigma}$.
2. Apply Algorithm 1—JOIN(Ω_{RC} , P, $\mid \sigma \mid$).
3. For $l = r + 1,\ldots, k$, set $e_{rl} = 1$ if $P(Rb_l) = Rb_r$.
4. If $r \leqslant k$, set $L_r = |\{l \mid P(Rl) = Rb_r\}|$.
   If $r = k + 1$, set $L_r = |\{l \mid P(Cl) = C1\}|$.
5. If $r = k + 1$, for each of the 2^{N-2} subsets T of $\{2,\ldots, N-1\}$:
   (a) Let τ denote $(\prod_{l\in T} \tau_l)\, \tau_N$ $(\tau_l = \mid \sigma_l \mid)$.
   (b) If $Rb_l{}^{\tau} = Rb_l$ for $l = k + 1,\ldots, n$, apply Algorithm 1—JOIN(Ω_C , P_0 , τ_C)—and go immediately to step 6 (disregarding any remaining subsets T).
6. Go back to step C.7.

As the algorithm progresses, $\{\sigma_1,\ldots, \sigma_N\}$ is the initial segment of the canonical generating set found thus far. Let K denote $\langle \sigma_1,\ldots, \sigma_N \rangle$, and let K_0 denote $K \cap R_0(H)$. At any time, $P(Ri)$ and $P(Cj)$ are the first points of the $\overline{K}$-orbits of Ri and Cj, respectively, $P_0(Cj)$ is the first point of the $\overline{K}_0$-orbit of Cj, L_i is the length of the ith basic orbit $Rb_i{}^K$ ($C1^K$ if $i = k + 1$), and e_{uv} is 1 if Rb_v is in the uth basic orbit and 0 otherwise $(1 \leqslant u < v \leqslant k)$.

Part A gives K, P, P_0 , $\{L_u\}$, and $\{e_{uv}\}$ their initial values, specifically, their values for the group $\langle -\iota \rangle$. An integer r, described below, is also initialized.

Parts B and C are a backtrack search for a next canonical generator $\sigma = \mu\tau$. In part B, $Ra_1,\ldots, Ra_i$ denote the images of $Rb_1,\ldots, Rb_k$ under the hypothetical τ, and r is minimal such that $a_r \neq b_r$. At this stage, we know that $\overline{K}^{(l)} = \overline{\mathrm{AUT}(H)}^{(l)}$ for $l > r$. From Propositions 4.1(a,b), 3.1, and 3.2,

[2] Recall that, for the second and subsequent applications of COLPERM, steps 2, 3, and 4 of COLPERM need not be repeated.

we know that $a_1,\ldots, a_i$ must satisfy the conditions (α), (β), (γ), and (δ) of step B.7.

When i reaches k and $a_1,\ldots, a_k$ satisfies (α)–(δ), control passes to part C. By Proposition 4.1(c) with $\Gamma = \{Rb_1,\ldots, Rb_n\}$, we know that $C1^\tau$ must be first in its $\overline{K}_0$-orbit. For each Cj first in its $\overline{K}_0$-orbit, the possibility $C1^\tau = Cj$ is checked (except, when $r = k+1$, the case $Cj = C1$ is omitted to avoid the identity permutation). In step C.3, the sign changes $\eta_1,\ldots, \eta_k$ for σ on $Rb_1,\ldots, Rb_k$ are computed. Steps C.4 and C.5 determine if there exists σ in AUT(H) (unique by Proposition 4.3) with $Rb_l^\sigma = \eta_l Ra_l$ for $l = 1,\ldots, k$. If σ exists, it is added to the canonical generating set and, if $r \leqslant k$, i is reset to r before resuming the search for another canonical generator $\sigma' = \mu'\tau'$; the reason for resetting i is that τ cannot agree with τ on $\{Rb_1,\ldots, Rb_r\}$; if it did, $\tau\tau^{-1} \in \overline{\mathrm{AUT}(H)}^{(r+1)} = \overline{K}^{(r+1)}$ and $\tau' \in \langle \tau, \overline{K}\rangle$. If σ does not exist, the search resumes just as if one of the tests (α)–(δ) had failed.

Part D adds to the canonical generating set and adjusts P, P_0, $\{L_u\}$, and $\{e_{uv}\}$ to correspond to the enlarged group $\overline{K}$; note L_u and e_{uv} change only when $u = r$. Step D.5, performed only when $r = k+1$, checks each new element of $\overline{K}^{(k+1)}$ (elementary Abelian by Corollary 4.5); if an element of $\overline{K}_0$ is found, that element together with the old $\overline{K}_0$ generate the new $\overline{K}_0$; the array P_0 is adjusted accordingly.

Several steps in the algorithm are optional in the sense that omitting them, partially or completely, affects only the efficiency of the algorithm and not its validity. Step D.5 and all parts of step B.7 are optional. Omitting E.5 or (α) or (β) of B.7 is not recommended as these steps are fast and reduce running times drastically if AUT(H) turns out to be large. However, it may not be best to perform (γ) and (δ) for every value of u, v, and w indicated. If (δ) is omitted entirely, Φ may be computed as needed in (γ). Otherwise, it is probably best to compute Φ and Ψ initially for all possible arguments in ascending order (note that the values of Φ and Ψ are independent of the order of the arguments).

By slight modification of step D.5, a minimal generating set for $R_0(H)$ may be obtained. Alternatively, using the fact that an element of $R_0(H)$ is determined uniquely by the image of a single column (Lemma 2.1), one can construct a short algorithm for computing $R_0(H)$. Replacing H by H^T, the same algorithm can be used to compute $C_0(H)$.

7. Determining Equivalence of Hadamard Matrices

Let H_1 and H_2 be two n-dimensional Hadamard matrices. With some modifications, the main algorithm may be used to determine if H_1 and H_2 are equivalent and, if so, to find $\sigma \in \hat{\Sigma}(\Omega_{RC})$ with $H_1^\sigma = H_2$.

We assume $b_1,\ldots, b_k$ is a common row base for H_1 and H_2 (rows of H_2 may

be permuted in order to minimize the size of a common row base). Using the main algorithm, we compute $\mathrm{AUT}(H_1)$ and $\mathrm{AUT}(H_2)$. If these groups have different orders, of course, H_1 and H_2 are not equivalent; the same applies if the two groups have different orbit lengths on Ω_{RC} (computable from P_1 and P_2). If $\sigma = \mu\tau$ is one isomorphism from H_1 to H_2, the set of such isomorphisms is the double coset $\mathrm{AUT}(H_1)\sigma\,\mathrm{AUT}(H_2)$. Thus we may assume τ is first in its right coset $\overline{\mathrm{AUT}(H_1)}\tau$ and in its left coset $\tau\,\overline{\mathrm{AUT}(H_2)}$ and that $C1^{\sigma} \in \Omega_C$ (not $-\Omega_C$). The main algorithm is modified as follows:

(1) Steps A.1 through A.5 are omitted; instead, $\{e_{uv}\}$ is given the values it had when the computation of $\mathrm{AUT}(H_1)$ terminated, and P and P_0 are given the values they had when the computation of $\mathrm{AUT}(H_2)$ terminated; r is set to 1.

(2) If the algorithm terminates at step B.4, then H_1 and H_2 are not equivalent.

(3) Each reference to H in the algorithm is replaced by a reference to H_1 or H_2 as follows:

(i) In B.7, each H on the left of an inequality becomes H_2 and each H on the right becomes H_1.

(ii) In C.3, C.4, and C.5, the first H becomes H_1 and the second becomes H_2.

(4) In C.1, j is always set to 1.

(5) Step C.7 and part D are omitted; step C.6 terminates the algorithm with σ the isomorphism from H_1 to H_2.

(6) Two additional conditions, denoted by (ϵ) and (ζ), are added to the four conditions in B.7:

(ϵ) The $\mathrm{AUT}(H_2)$ orbit of a_i has different length than the $\mathrm{AUT}(H_1)$ orbit of b_i (determinable from P_1 and P_2),

(ζ) For some j with $1 \leqslant j < i$, either $P_1(b_i) = b_j$ and $P_2(a_i) \neq P_2(a_j)$ or else $P_1(b_i) \neq b_j$ and $P_2(a_i) = P_2(a_j)$.

Alternatively, the computation of either $\mathrm{AUT}(H_1)$ or $\mathrm{AUT}(H_2)$ or both may be omitted; however, if $\mathrm{AUT}(H_1)$ is not computed, the test in B.7(β) must be eliminated, and if $\mathrm{AUT}(H_2)$ is not computed, the tests of B.7(α) and C.2 must be eliminated.

APPENDIX

Let H be the 4-dimensional Hadamard matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & - & - \\ 1 & - & 1 & - \\ 1 & - & - & 1 \end{bmatrix}.$$

Then $\{1, 2, 3\}$ is a row base for H; set $k = 3$, $b_1 = 1$, $b_2 = 2$, and $b_3 = 3$. Set $b_4 = 4$.

The output from the main algorithm is illustrated below:

$L_1 = 4$, $L_2 = 3$, $L_3 = 2$, $L_4 = 4$.
$N = 6$.

b	$R1$	$R2$	$R3$	$R4$	$C1$	$C2$	$C3$	$C4$
b^{σ_1}	$-R1$	$-R2$	$-R3$	$-R4$	$-C1$	$-C2$	$-C3$	$-C4$
b^{σ_2}	$R1$	$R2$	$-R3$	$-R4$	$C2$	$C1$	$C4$	$C3$
b^{σ_3}	$R1$	$-R2$	$R3$	$-R4$	$C3$	$C4$	$C1$	$C2$
b^{σ_4}	$R1$	$R2$	$R4$	$R3$	$C1$	$C2$	$C4$	$C3$
b^{σ_5}	$R1$	$R3$	$R2$	$R4$	$C1$	$C3$	$C2$	$C4$
b^{σ_6}	$R2$	$R1$	$R3$	$R4$	$C1$	$C2$	$-C4$	$-C3$
$P(b)$	$R1$	$R1$	$R1$	$R1$	$C1$	$C1$	$C1$	$C1$

REFERENCES

1. M. HALL, JR., "Combinatorial Theory," Blaisdell, Waltham, Mass., 1967.
2. D. E. KNUTH, Estimating the efficiency of backtrack programs, *Math. Comp.* **29** (1975), 121–136.
3. C. C. SIMS, Determining the conjugacy classes of a permutation group, *in* "Proceedings of the Symposium on Computers in Algebra and Number Theory," Amer. Math. Soc., New York, 1970.
4. C. C. SIMS, Computation with permutation groups, *in* "Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation," (S. R. Petrick, Ed.), Assoc. Comput. Mach., New York, 1971.