An Enumeration of Non-isomorphic One-factorizations and Howell Designs for the Graph K_{10} minus a One-factor

E. Seah and D.R. Stinson

ABSTRACT

We enumerate the (non-isomorphic) one-factorizations and sets of orthogonal one-factorizations of the graph $K_{10} - f$, where f is a one-factor of K_{10}. We find that there are 3192 one-factorizations; 18220 pairs, 3 triples, and 1 quadruple of mutually orthogonal one-factorizations.

1. Introduction.

Let Gr be an r-regular graph on n vertices. A *one-factorization* of Gr is a partition of the edge-set of Gr into r *one-factors,* each of which contains $n/2$ edges that partition the vertex set of Gr. Two one-factorizations F and G of Gr are *orthogonal* if any two edges of the graph which belong to the same one-factor of G belong to different one-factors of F (and vice-versa).

A *Howell Design* $H(s,t)$ is a square array of side s having the following properties: (1) each cell of the array is either empty or contains a two-subset of a t-set, (2) each element of the t-set occurs in exactly one cell of each row and each column, (3) any two-subset occurs in at most one cell of the array. It is easy to see that two orthogonal one-factorizations of Gr, an r-regular graph on n vertices, give rise to an $H(r,n)$; and, conversely, the existence of an $H(r,n)$ implies the existence of a pair of orthogonal one-factorizations of some r-regular graph on n vertices, Gr, which we call the *underlying graph* of the Howell Design.

In this paper, we enumerate the non-isomorphic one-factorizations, and sets of mutually orthogonal one-factorizations, of the graph $K_{10} - f$, where f is a one-factor. In particular, we enumerate all non-isomorphic $H(8,10)$'s, since the underlying graph of an $H(8,10)$ is $K_{10} - f$.

Denote $N(Gr) =$ the number of non-isomorphic one-factorizations of a graph Gr, and $N_i(Gr) =$ the number of non-isomorphic sets of i

ARS COMBINATORIA 21(1986), pp. 145-161

mutually orthogonal one-factorizations of Gr. In this paper, we prove that $N(K_{10}-f) = 3192$, $N_2(K_{10}-f) = 18220$, $N_3(K_{10}-f) = 3$ and $N_4(K_{10}-f) = 1$ (and $N_5(K_{10}-f) = 0$).

There results are interesting for several reasons. First, the non-isomorphic one-factorizations and Howell designs have been enumerated for all graphs on at most 10 vertices except $K_{10} - f$ (see [6]). Hence, the results of this paper complete this census. Also, the graph $K_{10} - f$ is the smallest graph (other than complete or complete bipartite graphs) for which there exists three (or more) orthogonal one-factorizations.

It has been conjectured that the maximum number of mutually orthogonal one-factorizations of a (regular) graph on n vertices is at most $(n-2)/2$. There are in fact infinitely many graphs for which (at least) $(n-2)/2$ mutually orthogonal one-factorizations are known to exist, but there are (obviously) no graphs known for which this conjectured bound is exceeded. The following results were previously known.

Theorem 1.1. *The following graphs have* at least $(n-2)/2$ *orthogonal one-factorizations:*

1) K_n, *if* $n - 1$ *is a prime power* $\equiv 3 \pmod 4$, *or* $n = 10$.

2) $K_{n/2,n/2}$, *if* $n/2$ *is a prime power.*

3) K_n *minus a one-factor, if* $n = 2^j + 2$, $j \geq 2$.

Proof. 1) is proved in [1] and [3]. The one-factorizations of the graphs in 2) are equivalent to mutually orthogonal Latin squares, so this result is well-known. The result 3) is proved in [4].

Hence, the four orthogonal one-factorizations of $K_{10} - f$ were previously known to exist. What we have done is to show that this set of four is unique, and that there is no set of five mutually orthogonal one-factorizations. Hence the graph $K_{10} - f$ provides another example of a graph which meets, but does not exceed, the bound. Thus it provides a little more empirical evidence in favour of this conjecture.

Also of interest are the algorithms used to establish the results of this paper. Our basic method is an *orderly* algorithm: we construct only non-isomorphic one-factorizations, by eliminating isomorphic structures as they are being constructed. These algorithms are described in the remainder of the paper. For those interested in orderly algorithms, we recommend [5].

2. An orderly algorithm for enumerating one-factorizations of a complete graph.

In this section, we outline an orderly algorithm that can be used to generate all the (non-isomorphic) one-factorizations of a complete graph

K_{2n}. We first need to define orderings on edges, one-factors, etc, of K_{2n}. All orderings are defined lexicographically, as follows:

- Suppose the vertices are numbered $1,\ldots,2n$. An edge e will be written as an ordered pair (p,p') with $1 \le p < p' \le 2n$.

- For any two edges $e_1 = (p_1,p'_1)$ and $e_2 = (p_2,p'_2)$, we say $e_1 < e_2$ if either of the following is true: (1) $p_1 < p_2$, (2) $p_1 = p_2$ and $p_1 < p'_2$.

- A one-factor f is a set of ordered edges, i.e. $f = (e_1,e_2,e_3,\ldots,e_n)$, where $e_1 < e_2 < e_3 < \cdots < e_n$.

- For two one-factors $f_i = (e_{i1},e_{i2},e_{i3},\ldots,e_{in})$ and $f_j = (e_{j1},e_{j2},e_{j3},\ldots,e_{jn})$, we say $f_i < f_j$ if there exists a k $(1 \le k \le n)$ such that $e_{il} = e_{jl}$ for all $l < k$, and $e_{ik} < e_{jk}$.

- A one-factorization F of K_{2n} is an ordered set of $2n-1$ one-factors, i.e. $F = (f_1,f_2,\ldots,f_{2n-1})$, where $f_i < f_j$ whenever $i < j$. We use F, G, H to denote one-factorizations, and f_i, g_i, h_i the corresponding one-factors.

- For two one-factorizations F and G, we say that $F < G$ if there exists some i, $1 \le i \le 2n-1$, such that $f_i < g_i$, and $f_j = g_j$ for all $j < i$.

- For $1 \le i \le 2n-1$, $F_i = (f_1,f_2,\ldots,f_i)$ will denote a partial one-factorization consisting of an ordered set of i one-factors. We say that i is the *rank* of the partial one-factorization. Note that $F_{2n-1} = F$, a (complete) one-factorizations. We can also extend our ordering to partial one-factorizations of rank i, in an analogous manner.

Given a partial one-factorization F_i (of rank i), we can rename the $2n$ points using a permutation α and obtain another partial one-factorization, denoted F_i^α. We say F_i is *canonical* if $F_i^\alpha \ge F_i$ for all permutations α. It is easy to see that if two partial one-factorizations of rank i, F_i and G_i, are distinct and are both canonical, then F_i and G_i are non-isomorphic. Also, if $F_i = (f_1,f_2,\ldots,f_i)$ is canonical, and $1 \le j \le i$, then $F_j = (f_1,f_2,\ldots,f_j)$ is also canonical.

Let $\mathbf{F}_i$ denote the set of canonical partial one-factorizations of rank i. An orderly algorithm will generate each set $\mathbf{F}_i$ of canonical partial one-factorizations of rank i in turn, starting with $i = 1$ and ending with $i = 2n-1$. Once the whole process is through, $\mathbf{F}_{2n-1}$ is the set of all the non-isomorphic one-factorizations of K_{2n} (in canonical form).

Define $\mathbf{S}_i$ to be the set of all one-factors containing the edge $(1,i+1)$. It is easy to see that any $F_i \in \mathbf{F}_i$ must contain one one-factor from each of $\mathbf{S}_1, \ldots, \mathbf{S}_i$. The following pseudo-code describes how to generate $\mathbf{F}_{i+1}$ from $\mathbf{F}_i$ (step $i+1$):

$\mathbf{F}_{i+1} = \emptyset$;

For each $F_i \in \mathbf{F}_i$ do

For each one-factor $f \in \mathbf{S}_{i+1}$ that is disjoint from all one-factors of F_i do

For each permutation α do

(1) compute f^α and F_i^α;

(2) if $F_i^\alpha \cup \{f^\alpha\} < F_i \cup \{f\}$ then $F_i \cup \{f\}$ is not canonical, so discard it and go on to next f

{Here $F_i^\alpha \cup \{f^\alpha\} \geq F_i \cup \{f\}$ for all α. Hence $F_i \cup \{f\}$ is canonical, so save it for the next step.}

$\mathbf{F}_{i+1} = \mathbf{F}_{i+1} \cup \{F_i \cup \{f\}\}$.

We begin the algorithm by describing $\mathbf{F}_1$. $\mathbf{F}_1$ has only one element, namely $f_a = ((1\ 2)(3\ 4)(5\ 6)...(2n-1\ 2n))$, the very first one-factor in S_1, as all other one-factors in S_1 can be mapped into f_a (and hence are isomorphic).

3. One-factorizations of K_{10}.

As we have described it, we would try all $(2n)!$ permutations of $\{1,...,2n\}$ as our α's. This is a lot of work, even for small values of n (eg, if $n = 5$, then $(2n)! = 10! = 3628800$). However, we can do a lot better than this.

We are interested only if those α's that cause $F_i^\alpha < F_i$. It is necessary only to try those α's that map a one-factor of $F_i \cup \{f\}(f \in \mathrm{S}_{i+1})$ into f_a. The number of such mappings is $(i+1)\cdot 2^n \cdot n!$. For $n = 5$, the maximum number would be $(8+1)\cdot 3840 = 34560$, which is only 1/105 of all the $(2n)!$ permutations.

A further improvement can be achieved by testing only those α's which map two one-factors of $F_i \cup \{f\}$ into a fixed set of two one-factors, which is the approach we use. We observe that any two disjoint one-factors of K_{10} form either two disjoint cycles of length 4 and 6 (type '46') of a Hamiltonian circuit of length 10 (type '10'). The smallest one-factor in S_2 that forms a type '46' structure with $f_a = ((1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10))$ is $f_b = ((1\ 3)(2\ 4)(5\ 7)(6\ 9)(8\ 10))$, and the smallest that forms a type '10' is $f_c = ((1\ 3)(2\ 5)(4\ 7)(6\ 9)(8\ 10))$. It follows then that $\mathbf{F}_2 = \{(f_a,b_b),(f_a,f_c)\}$, where $f_a < f_b < f_c$.

To see how we map two one-factors of $F_i \cup \{g\}(= (f_1,f_2,...,f_{i+1}))$ at step $i + 1$ into two one-factors, we consider the following two cases:

(1) $f_1 f_2 = f_a f_b$ (type '46'):
We map any $f_j f_k$, $1 \leq j < k \leq i+1$ of type '46' into $f_a f_b$ (in such a way that f_j or f_k is mapped to f_a). To map into *any other* two

one-factors of type '46' would always make $F_i^\alpha > F_i$. There are $2 \cdot (2 \cdot 2) \cdot (2 \cdot 3) = 48$ ways to do this.

We may ignore those $f_j f_k$ of type '10', as mapping them into $f_a f_c$ would always make $F_i^\alpha > F_i$. (In general, if $f_1 f_2$ is of type x, we may ignore $f_j f_k$ of type y so long as the canonical two one-factors corresponding to type y are greater than those of type x.) The maximum number of mappings α required in this case is $(9 \cdot 8)/2 \cdot 48 = 1728$, which is 20 times better than mapping one-factor to another.

(2) $f_1 f_2 = f_a f_c$ (type '10'):
All $f_j f_k$, $1 \leq j < k \leq i+1$ must be of type corresponding to a canonical structure less than $f_1 f_2$). Thus we discard those $g \in S_{i+1}$ which form a type '46' structure with any of f_j, $1 \leq j \leq i$, before the canonicity testing. There are $2 \cdot (2 \cdot 5) = 20$ ways to map type '10' structures. The maximum number of such mappings is $(9 \cdot 8)/2 \cdot 20 = 720$.

Table 1 gives the number of canonical structures and CPU time taken for each of the steps. The number of (complete) one-factorizations of K_{10} agrees with the results in Gelling [2]. The table shows that the number of canonical structures increases steadily during the earlier steps, then decreases at a slower pace in the later steps. All the computer work in this paper is implemented in Pascal/VS and run on the University of Manitoba Amdahl 580 computer.

Table 1
Non-isomorphic Canonical Partial One-factorization of K_{10}

Step	# of canonical structures at step $i+1$			CPU time
$i+1$	type '46'	type'10'	total	(in seconds)
3	6	6	12	1
4	80	21	101	3
5	586	24	610	20
6	1608	14	1622	89
7	1722	9	1731	181
8	819	1	820	186
9	395	1	396	147
				627

4. One-factorizations of $K_{10} - f$.

Without loss of generality, we let $f = f_a$. The algorithm is very similar to that of K_{10}. The following differences are noted:

(1) The one-factorizations of $K_{10} - f$ have 8 one-factors and do not include the five edges in f_a.

(2) We pretend that f_a is part of the one-factorization of $K_{10} - f$. That is, we start out with $\mathbf{F}_1 = \{f_a\}$, and go through the steps as in the case of K_{10}. We can ignore f_a after $\mathbf{F}_9$ is produced.

(3) In testing whether $\mathbf{F}_{i+1} = \mathbf{F}_i \cup \{g\}$, $g \in \mathbf{S}_{i+1}$ is canonical, we observe that

(a) $f_1^\alpha = f_1\ (= f_a)$.

(b) We will map two one-factors into two one-factors, except we need only examine $f_1 f_j$, $j > 1$.

(c) In the case of $f_1 f_2 = f_a f_b$ (type '46'), we ignore $f_1 f_j$ of type '10', as in K_{10}. There are precisely 24 ways (one half of 48) that $f_1 f_j$ of type '46' can be mapped into $f_a f_b$ such that f_1 is fixed. The maximum number of mappings for an F_{i+1} is $(i+1)\cdot 24$.

(d) In the case of $f_1 f_2 = f_a f_c$ (type '10'), all $f_1 f_j$, $j > 1$ must be of type '10', while $f_k f_j$, $j,k \neq 1$ can be of either type. Again the number of ways that $f_1 f_j$ can be mapped into $f_a f_c$ is reduced by half to 10. The number of mappings for a F_{i+1} is $(i+1)\cdot 20$.

The number of canonical structures and CPU time required for each of the steps are listed in Table 2. The number of non-isomorphic one-factorizations of $K_{10} - f$ of types '46' and '10' are 2944 and 248 respectively. The algorithm required approximately 18 minutes of CPU time.

Table 2
Non-isomorphic Canonical Partial One-factorizations of $K_{10} - f$

Step $i+1$	# of canonical structures at step $i+1$ type '46'	type '10'	total	CPU time (in seconds)
3	7	15	22	1
4	114	109	223	2
5	1039	412	1451	12
6	4600	1136	5736	67
7	7802	1437	9239	206
8	4917	610	5527	385
9	2944	248	3192	401
				1074

5. Howell designs $H(8,10)$.

In enumerating the non-isomorphic pairs of orthogonal one-factorizations of $K_{10} - f$, we extend the canonicity concept as follows:

- A set of two orthogonal one-factorizations F and G are written as an ordered pair (F,G), with $F < G$. As in the case of $K_{10} - f$, we pretend that f_a is part of the one-factorizations. Denote $F = (f_1, f_2, \ldots, f_9)$, $G = (g_1, g_2, \ldots, g_9)$, where $f_1 = g_1 = f_a$.
- We say (F,G) is canonical if, for all α's that fix f_a, $(F,G)^\alpha \geq (F,G)$.

We have the following two observations:

(1) If (F,G) is canonical, F is necessarily canonical. Otherwise we can find an α such that $F^\alpha < F$ and make $(F,G)^\alpha < (F,G)$.

(2) Two distinct, canonical (F,G) are non-isomorphic.

Hence, in generating pairs of orthogonal one-factorizations, we can take, in turn, each (canonical) one-factorization F of $K^{10} - f$ produced in the previous section, and generate all G's that are orthogonal to and greater than F.

However, it is easy to see that a given (F,G), where $F < G$ and F is canonical, is not necessarily canonical. In testing whether (F,G) is canonical, we need to check all α's that make F^α canonical and all α's that make G^α canonical:

(1) α's for F: It suffices to examine those α's such that $F^\alpha = F$, since F is canonical. That is, we can restrict the α's to the automorphism group of F. If, for any such α, $G^\alpha < G$, then (F,G) is not canonical. Note that if $f_1 f_2 = f_a f_c$ (hence all $f_1 f_j$ are of type '10'), then all $g_1 g_j$ must necessarily be of type '10'.

(2) α's for G. There are two cases:

(a) There exists a g_1g_j of type '46': we map all g_1g_j of type '46' into f_af_b, and ignore those g_1g_j of type '10'.

(b) All g_1g_j are of type '10': we map them into f_af_c.

Using these permutations α (for G), there are three situations where (F,G) is not canonical, as described by the following pseudo-code:

If $G^\alpha < F$ then (F,G) is not canonical

Else

If $G^\alpha = F$ then

If $F^\alpha < G$ then (F,G) is not canonical

Else

If $(F^\alpha = F)$ and $(G^\alpha < G)$ then (F,G) is not canonical.

We now outline the algorithm that we use to generate all the non-isomorphic Howell designs $H(8,10)$:

For each F in the set $\mathbf{F}$ of non-isomorphic one-factorizations of $K_{10} - f$ do:

1. Generate from $\mathbf{S}_i$, $i = 2,\ldots,8$, the set $\mathbf{T}$ of one-factors that intersect each of the one-factors of F in at most one edge.
2. Construct all possible one-factorizations G, which consist only of one-factors from $\mathbf{T}$, discarding those G's $< F$. These G's are all orthogonal to F. Note that $g_1 = f_a$, where $G = (g_1,g_2,\ldots,g_9)$.
3. If no G's were constructed in step 2, then go on to next F.
4. Determine the automorphism group $A = \{\alpha\colon F^\alpha = F\}$ of F.
5. For each G do:

   (a) map g_1g_j into f_af_b or f_af_c as described earlier. If $(F,G)^\alpha \geq (F,G)$ for all α's, proceed to (b); otherwise (F,G) is not canonical, so go on to next F.

   (b) apply each $\alpha \in A$ to G. If for all $\alpha \in A$, $G^\alpha \geq G$, then (F,G) is canonical; otherwise it is not.

In total, the number of non-isomorphic (F,G) of $K^{10} - f$ generated is 18220. It required 38 minutes of CPU time. Table 3 in the Appendix gives the frequency distribution of these designs, based on the number of non-isomorphic (F,G) (where $F < G$) for a given F. It is interesting to observe the wide variation in the numbers of orthogonal mates. 540 one-factorizations F had no orthogonal mates $G > F$, while, at the other extreme, one of the one-factorizations had 63 orthogonal mates.

6. Howell cubes and $H_4(8,10)$.

We write a set of three mutually orthogonal one-factorizations of $K_{10} - f$ as an ordered triplet (F,G,H) with $F < G < H$. We say that (F,G,H) is canonical if $(F,G,H)^\alpha \geq (F,G,H)$ for all α's that fix f_a.

For (F,G,H) to be canonical, F is necessarily canonical, and so is (F,G). These observations suggest the following algorithm:

For each non-isomorphic F of $K_{10} - f$ do:

1. Construct from **T** all possible one-factorizations G with $F < G$, as in steps 1 and 2 in the previous algorithm.
2. Examine all pairs of one-factorizations G,H where G and H are constructed in step 1. If G and H are orthogonal, then we have a set (F,G,H) of three mutually orthogonal one-factorizations.
3. Determine which triples (F,G,H) are canonical.

In total, we find 12 triples (F,G,H) in step 2. We immediately eliminate 7 of them, as their corresponding (F,G)'s are not canonical. The first (smallest) set is necessarily canonical (set 1 in Table 4). Three of the 12 sets, which are all distinct from set 1, form a quadruple (F,G,H,I); hence the corresponding (F,G,H) must be canonical (set 3 in Table 4). This leaves us with 3 sets to which we apply canonicity testing (in this case we simply try all α's that map f_a into f_a); we find one of them is canonical (set 2 in Table 4). In summary, we have

1. $N_3(K_{10}-f) = 3$. The corresponding Howell cubes are shown in Table 4.
2. $N_4(K_{10}-f) = 1$. Table 5 gives the corresponding $H_4(8,10)$.

It is interesting to note that the set of four mutually orthogonal one-factorizations can be constructed from a finite projective plane of order 8 [4].

We present the automorphism groups A of the non-isomorphic Howell cubes and $H_4(8,10)$ in Table 6.

7. Summary.

We describe an orderly algorithm that we use to determine the one-factorizations and sets of orthogonal one-factorizations of the graph $K_{10} - f$, where f is a one-factor of K_{10}. There are 3192 one-factorizations; 18220 pairs, 3 triples, and 1 quadruple of mutually orthogonal one-factorizations.

References.

[1] J.H. Dinitz and W.D. Wallis, *Four orthogonal one-factorizations on ten points,* in "Algorithms in Combinatorial Design Theory", Annals of Discrete Math. 26 (1985), 143-150.

[2] E.N. Gelling, *On one-factorizations of the complete graph and the relationship to round robin schedules,* M. Sc. Thesis, University of Victoria, 1973.

[3] K.B. Gross, R.C. Mullin and W.D. Wallis, *On the number of pairwise orthogonal symmetric Latin squares,* Utilitas Math. 4 (1973) 239-251.

[4] E.R. Lamken and S.A. Vanstone, *Designs with mutually orthogonal resolutions,* preprint.

[5] R.C. Read, *Every one a winner,* Annals of Discrete Math. 2 (1978) 107-120.

[6] A. Rosa and D.R. Stinson, *One-factorizations of regular graphs and Howell designs of small order,* Utilitas Math., to appear.

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
R3T 2N2 Canada

Table 3
Frequency Distribution of Non-isomorphic sets of Two Mutually Orthogonal one-factorizations of $K_{10} - f$

j	$Fr(j)$	$j^*Fr(j)$
0	540	0
1	373	373
2	301	602
3	286	858
4	268	1072
5	220	1100
6	191	1146
7	153	1071
8	135	1080
9	109	981
10	88	880
11	81	891
12	75	900
13	48	624
14	52	728
15	34	510
16	38	608
17	27	459
18	20	360
19	18	342
20	17	340
21	10	210
22	10	220
23	10	230
24	18	432
25	11	275
26	5	130
27	8	216
28	9	252
29	4	116
30	8	240
31	4	124
32	1	32
35	3	105
36	1	36
37	1	37
38	3	114
39	3	117
40	1	40

41	1	41
42	1	42
43	1	43
44	2	88
45	1	45
47	1	47
63	1	63
	3192	18220

$Fr(j)$: Number of one-factorizations F for which the number of non-isomorphic canonical pairs of one-factorizations of the form (F,G) is j.

Table 4
Howell Cubes $H_3(8,10)$

Set 1

(F,G):

1	3					5	7	8	10	6	9	2	4		
7	9	1	4	6	10			2	3			5	8		
		8	9	1	5	4	10			2	7	3	6		
		3	5	2	8	1	6					7	10	4	9
		2	10	3	9			1	7	4	5			6	8
5	10					2	9	4	6	1	8			3	7
4	8	6	7							3	10	1	9	2	5
2	6			4	7	3	8	5	9					1	10

(F,H):

1	3	6	9	2	4					5	7	8	10		
		1	4	7	9	2	3	6	10					5	8
4	10			1	5	8	9			3	6			2	7
2	8	7	10			1	6	3	5	4	9				
				6	8			1	7	2	10	4	5	3	9
						5	10	2	9	1	8	3	7	4	6
6	7	2	5	3	10			4	8			1	9		
5	9	3	8			4	7					2	6	1	10

(G,H):

1	3			7	9	5	10	4	8			2	6		
6	7	1	4			8	9	3	5	2	10				
2	8			1	5	4	7	6	10					3	9
4	10	3	8			1	6	2	9	5	7				
5	9					2	3	1	7			8	10	4	6
		6	9	3	10					1	8	4	5	2	7
		7	10	2	4					3	6	1	9	5	8
		2	5	6	8					4	9	3	7	1	10

Set 2

(F,G):

1	3			8	10	2	4			6	9			5	7
		1	4	2	3	7	9	5	10			6	8		
8	9	2	6	1	5					3	7	4	10		
2	5	7	10			1	6	4	8					3	9
6	10			4	9			1	7			3	5	2	8
		5	9			3	10			1	8	2	7	4	6
4	7					5	8	3	6	2	10	1	9		
		3	8	6	7			2	9	4	5			1	10

(F,H):

1	3			6	9			8	10			5	7	2	4
		1	4			5	10			7	9	2	3	6	8
		8	9	1	5			2	6	4	10			3	7
7	10					1	6	3	9	2	5	4	8		
		3	5	2	8	4	9	1	7			6	10		
4	6			3	10	2	7			1	8			5	9
5	8	2	10	4	7					3	6	1	9		
2	9	6	7			3	8	4	5					1	10

(G,H):

1	3	8	9	4	7					2	5	6	10		
7	10	1	4			3	8	2	6					5	9
		6	7	1	5	4	9	8	10			2	3		
5	8			3	10	1	6			7	9			2	4
2	9					5	10	1	7	3	6	4	8		
		2	10	6	9			4	5	1	8			3	7
		3	5			2	7			4	10	1	9	6	8
4	6			2	8			3	9			5	7	1	10

Set 3

(F,G):

1	3	5	7			8	10			6	9			2	4
6	8	1	4	7	9			5	10			2	3		
4	9			1	5					2	7	6	10	3	8
		3	10			1	6	2	8			4	7	5	9
				3	6	2	9	1	7	4	10	5	8		
				2	10	4	5	3	9	1	8			6	7
7	10	2	6	4	8					3	5	1	9		
2	5	8	9			3	7	4	6					1	10

(F,H):

1	3	8	10			2	4	6	9			5	7		
7	9	1	4	2	3					5	10			6	8
6	10	2	7	1	5					4	9	3	8		
2	8	5	9			1	6	3	10					4	7
		3	6	4	10	5	8	1	7					2	9
4	5			6	7	3	9			1	8	2	10		
						7	10	4	8	2	6	1	9	3	5
				8	9			2	5	3	7	4	6	1	10

(G,H):

1	3					7	10	2	5	4	9			6	8
		1	4	8	9			3	10	2	6	5	7		
7	9	3	6	1	5			4	8			2	10		
4	5	8	10			1	6			3	7			2	9
2	8					3	9	1	7	5	10	4	6		
		2	7	4	10		6	9	1	8			3	5	
6	10			2	3	5	8					1	9	4	7
		5	9	6	7	2	4					3	8	1	10

Table 5

$H_4(8,10)$

(F,G): see Table 4, Set 3

(F,H): see Table 4, Set 3

(G,H): see Table 4, Set 3

(F,I):

1	3	6	9	8	10			2	4					5	7
5	10	1	4			7	9			2	3	6	8		
				1	5	3	8	6	10			2	7	4	9
				4	7	1	6			5	9	3	10	2	8
2	9	5	8			4	10	1	7					3	6
6	7	2	10	3	9					1	8	4	5		
4	8			2	6			3	5	7	10	1	9		
		3	7			2	5	8	9	4	6			1	10

(G,I):

1	3					2	5			7	10	6	8	4	9
		1	4	2	6			8	9			3	10	5	7
4	8	2	10	1	5	7	9							3	6
2	9	3	7	8	10	1	6					4	5		
5	10			3	9			1	7	4	6			2	8
		6	9			4	10	3	5	1	8	2	7		
		5	8	4	7			6	10	2	3	1	9		
6	7					3	8	2	4	5	9			1	10

(H,I):

1	3					7	9	6	10			4	5	2	8
		1	4	8	10					5	9	2	7	3	6
6	7			1	5	4	10	8	9	2	3				
		5	8	3	9	1	6	2	4	7	10				
4	8	6	9			2	5	1	7			3	10		
5	10	3	7	2	6					1	8			4	9
		2	10			3	8			4	6	1	9	5	7
2	9			4	7			3	5			6	8	1	10

Table 6

Automorphism Groups of $H_3(8,10)$ and $H_4(8,10)$

$H_3(8,10) = (F,G,H)$

Set 1 $A = \langle I \rangle$.

Set 2 $A = \langle g \rangle \cong \mathbf{Z}_8$, where $g = (3\ 5\ 8\ 10\ 4\ 6\ 7\ 9)$.
A interchanges G and H.

Set 3 $A = \langle g \rangle \cong \mathbf{Z}_6$, where $g = (5\ 6)(3\ 8\ 10\ 4\ 7\ 9)$.
A maps F into G, G into H, and H into F.

$H_4(8,10) = (F,G,H,I)$

$A = \langle g1,g2 \rangle$, $|A| = 24$,
and $g1 = (3\ 4)(5\ 10\ 8\ 6\ 9\ 7)$.
$g2 = (5\ 6)(3\ 8\ 10\ 4\ 7\ 9)$.
$g1$ maps H into G, G into I, and I into H.
$g2$ maps F into G, G into H, and H into F.